AFRL-IF-RS-TR-2007-180
Final Technical Report
July 2007

DARPA QUANTUM NETWORK TESTBED

BBN Technologies
Sponsored by Defense Advanced Research Projects Agency
DARPA Order No. L750

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-IF-RS-TR-2007-180 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

D.J. NICHOLSON
Work Unit Manager

WARREN H. DEBANY, Jr.
Technical Advisor, Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUL 2007	Final	Mar 01 – Feb 07

4. TITLE AND SUBTITLE

DARPA QUANTUM NETWORK TESTBED

5a. CONTRACT NUMBER
F30602-01-C-0170

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62716E

6. AUTHOR(S)

Chip Elliott and Henry Yeh

5d. PROJECT NUMBER
L750

5e. TASK NUMBER
QN

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BBN Technologies
10 Moulton St.
Cambridge MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Dr.
Arlington VA 22203-1714

AFRL/IFGC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2007-180

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# AFRL-07-0057

13. SUPPLEMENTARY NOTES

14. ABSTRACT

BBN has designed and built the world's first Quantum Network testbed, delivering end-to-end network security via high-speed Quantum Key Distribution (QKD), and testing that Network against sophisticated eavesdropping attacks. BBN has fielded this ultra-high-security network into commercial fiber across the metro Boston area. BBN's QKD network comprises 10 nodes. It is both extremely secure and 100% compatible with today's Internet technology. Four of the 10 nodes are running 24x7 over Boston metro telecom fiber between BBN, BU and Harvard and protecting Internet traffic; four other nodes are free-space; and two are based on polarization entanglement through fiber. BBN also teamed with NIST & University of Rochester to build the first superconducting single-photon detector. It saw "first light" in 2005. We characterized our prototype detector at temperatures ranging from 2K to 4K and expect to operate a full detector suite in early 2006 at speed up to 100MHz (20x faster than any existing detector). BBN also collaborated with MIT to build the world's first experimental demonstration of Eve, a quantum eavesdropper. The results were published in Summer 2006.

15. SUBJECT TERMS
Quantum cryptography, key distribution, network security

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			WUM
U	U	U	UL	164	19b. TELEPHONE NUMBER (*Include area code*) N/A

TABLE OF CONTENTS

List of Figures

Program Name:	**Quantum Information Science and Technology (QuIST)**
Prism Ref Id:	12673
Contract #:	F30602-01-C-0170
Description:	Design and build the world's first Quantum Network test bed, delivering end-to-end network security via high-speed Quantum Key Distribution, and test that Network against sophisticated eavesdropping attacks
Customer/Cust#:	DARPA
POP:	08/10/2001 - 02/28/2007
Revenue Type:	CPFF

1 Program Plan and Goals

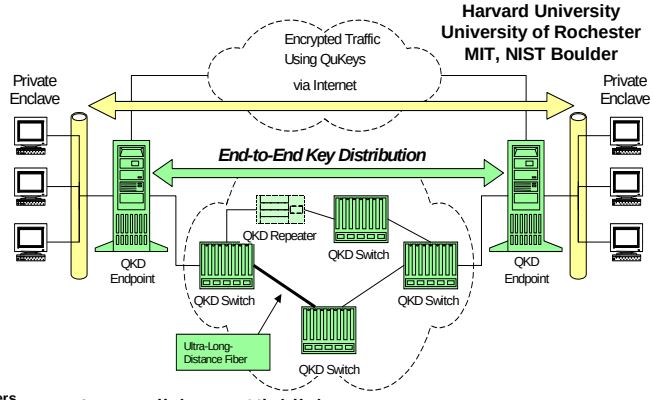
Building the DARPA Quantum Network

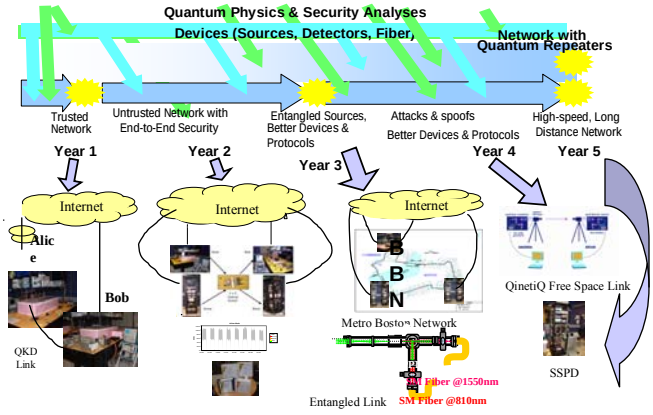
BBN TECHNOLOGIES

BU Photonics Center
Harvard University
University of Rochester
MIT, NIST Boulder

Objective:

BBN proposes to design and build the world's first Quantum Network, delivering end-to-end network security via high-speed Quantum Key Distribution (QKD), and testing that Network against sophisticated eavesdropping attacks.





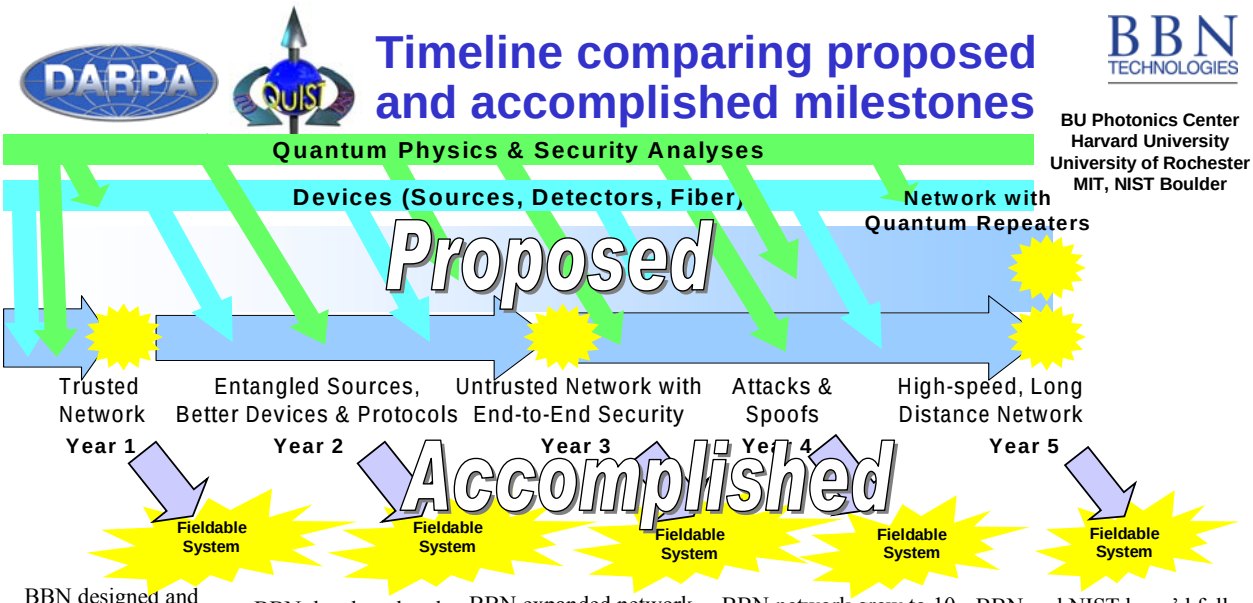
Accomplishment Highlights:

BBN's current QKD network consists 10 nodes. It is both extremely secure and 100% compatible with today's Internet technology. Four out of 10 nodes are running 24x7 over Boston metro telecom fiber between BBN, BU, and Harvard and protecting Internet traffic; 4 other nodes are free-space; and 2 are based on polarization entanglement through fiber.

BBN also teamed with NIST & University of Rochester to build the first superconducting single-photon detector. It saw "first light" in 2005. We characterized our prototype detector at temperatures ranging from 2 K to 4 K and expect to operate a full detector suite in early 2006 at speed up to 100 MHz (20x faster than any existing detector).

BBN also collaborated with MIT to build the world's first experimental demonstration of Eve, a quantum eavesdropper. The results were published in summer 2006.

2 Highlight Program Accomplishments



BBN designed and built a full telecom-compatible system from phase-modulated laser source, including an entire "industrial strength" QKD protocol stack, and continuous active adjustment of the interferometer arm lengths.

BBN developed and constructed a 4 node network of "Mark 2" QKD systems connected through an "untrusted" optical switch across metro Boston; it has run steadily since 6/2004, protecting Internet traffic with end-to-end security.

BBN expanded network to 8 heterogenous nodes, with BU entangled source and NIST free-space link. Designed and created key-relay for long distance quantum cryptography and/or air-fiber interfaces; running nonstop in metro Boston network.

BBN network grew to 10 nodes, adding QinetiQ-built free space link running BBN QKD protocols. Detailed security analyses reveal many types of attack; invented a new form of QKD error correction (Niagara); started work on ultrafast Superconducting Single-Photon Detector.

BBN and NISTdemo'd full SSPD clocked at 100 MHz, and incorporated into Boston-area Bob system (5 MHz). BU demonstrated first high-fidelity source of entangled photons at 1550nm; new source used with SSPD as heralded single photon source. MIT and BBN start building the world's first Eve (proof-of-concept).


Bulletized list of accomplishments (highlights)

BBN TECHNOLOGIES

BU Photonics Center
Harvard University
University of Rochester
MIT, NIST Boulder

- **Fielded the world's first quantum cryptography network, operating it continuously beneath the streets of Cambridge, Massachusetts since June 2004.** The DARPA Quantum Network links BBN's campus to Harvard University and the Boston University and have been operating 24x7 continuously since June 2004, and protecting Internet traffic between our campuses. The network contains 10 heterogenous nodes: 4 systems built for metro telecom fiber & connected through a functioning optical switch, 2 different free-space links, and 1 entangled link**.** All run BBN's industrial strength QKD software, and perform full-scale quantum cryptography.
- Implemented a full, "industrial strength" Quantum Key Distribution protocol stack including key-relay protocol, and operated it on a range of QKD nodes including on hardware built by NIST and by QinetiQ.
- Developed and implemented a highly secure network and **100% compatible with conventional Internet technology, including IPsec for IPv4 and IPv6**. BBN has designed and coded a much improved Quantum Perfect Forward Secrecy (QPFS) and Quantum One-Time-Pad (QOTP) over IPSec extensions.
- Invented and implemented 'Niagara,' a new form of **quantum error detection and correction** based on Low-Density Parity Check (LDPC) codes that provides highly efficient, one-pass operation near the Shannon Limit.
- Designed and built the **world's fastest single-photon detector for telecom wavelengths** – the Superconducting Single-Photon Detector (SSPD), in close collaboration with U. Rochester and NIST Boulder. Demo'd its operation at 100 MHz, 20x faster than current generation technology (Epitaxx APDs). This opens up a path towards ultra-fast QKD through fiber, e.g., at 10 GHz pulse rate and up.
- Demonstrated full-scale trial operation of quantum cryptography with a prototype SSPD detector, and performed an initial comparison of it with the APD through metro fiber. Will characterize SSPD at 100 MHz and perform bell-state measurements in coming months.
- Built and demonstrated the **world's first Type I and Type II sources of polarization-entangled photon pairs at 1550nm**.
- Built a highspeed (~10 MHz) **physical random number generator**, and integrated it into Bob. This design provides an upgrade path to ultra-highspeed random number generators (10 GHz and up) which are needed for fast quantum cryptography.
- Designed and implemented active Path Length Control that allow us to run a phase-modulated quantum cryptography network continuously without an interruption, instead of as a one-shot science experiment.
- Developed a full suite of architecture, design, and engineering documents, found at quantum.bbn.com
- Collaborated with OptiMetrics to publish a standard, open interface for QKD systems, based on working BBN QKD interface
- Suspected and then showed that space dimensions are important to quantum cryptography.
- By ignoring electron-photon interactions, modeled propagation of light in fiber and showed how accounting for entanglement both in frequency and in polarization enables more key to be distilled.

3 DARPA QuIST Team Members

PI: Chip Elliott	PM: Henry Yeh
BBN Technical staff	
Dave Pearson	Alex Colvin
John Lowry	John Schlafer
Greg Troxel	Oleksiy Pikalo
Jonathan Habif	Bill Nelson
Summer visiting students	
Rich Cannings	University of Calgary
Eric VanWyk	Olin College
Michael Yamartino	Brown University
Boston University Teammate	
Alexander Sergienko	Malvin Carl Teich

Bahaa Saleh	Gregg Jaeger
Martin Jaspan	Giovanni Di Giuseppe
Brian Imhausen	
Harvard University Teammate	
John Myers	Tai Wu
Leo Donnelly	
NIST Teammate	
Sae Woo Nam	Bob Schwall
Robert Hadfield	Aaron Miller
Alan Mink	Joshua Bienfang
Carl J. Williams	
MIT Teammate	
Franco Wong	Jeffrey Shapiro
University of Rochester Teammate	
Roman Sobolewski	Aaron Pearlman
Northwestern University Teammate	
Prem Kumar	Greg Kanter
Chuang Liang	

The rest of this document, Quantum network system and test bed, synthesizes the architecture, design, implementation, and illustrations of Rich Cannings, Alex Colvin, John Myers, David Pearson, Oleksiy Pikalo, John Schlafer, Greg Troxel, and Chip Elliott, with suggestions and comments by the entire DARPA Quantum Network team including Henry Yeh and John Lowry.

4 Quantum network system and Testbed Summary

BBN has designed and built the world's first Quantum Network testbed, delivering end-to-end network security via high-speed Quantum Key Distribution, and testing that Network against sophisticated eavesdropping attacks.

BBN has fielded this ultra-high-security network into commercial fiber across the metro Boston area. BBN's QKD network comprises 10 nodes. It is both extremely secure and 100% compatible with today's Internet technology. Four of the 10 nodes are running 24x7 over Boston metro telecom fiber between BBN, BU, and Harvard and protecting Internet traffic; 4 other nodes are free-space; and 2 are based on polarization entanglement through fiber.

BBN also teamed with NIST & University of Rochester to build the first superconducting single-photon detector. It saw "first light" in 2005. We characterized our prototype detector at temperatures ranging from 2 K to 4 K and expect to operate a full detector suite in early 2006 at speed up to 100 MHz (20x faster than any existing detector).

BBN also collaborated with MIT to build the world's first experimental demonstration of Eve, a quantum eavesdropper. The results were published in summer 2006.

The rest of this document describes the architecture, design and development of BBN's Quantum Network Testbed.

5 Scope

5.1 Identification

This document provides an overall description of what the system does, its major components, and how those components fit together. One may also refer to the separate document for each subsystem for a further level of details as regards the subsystems.

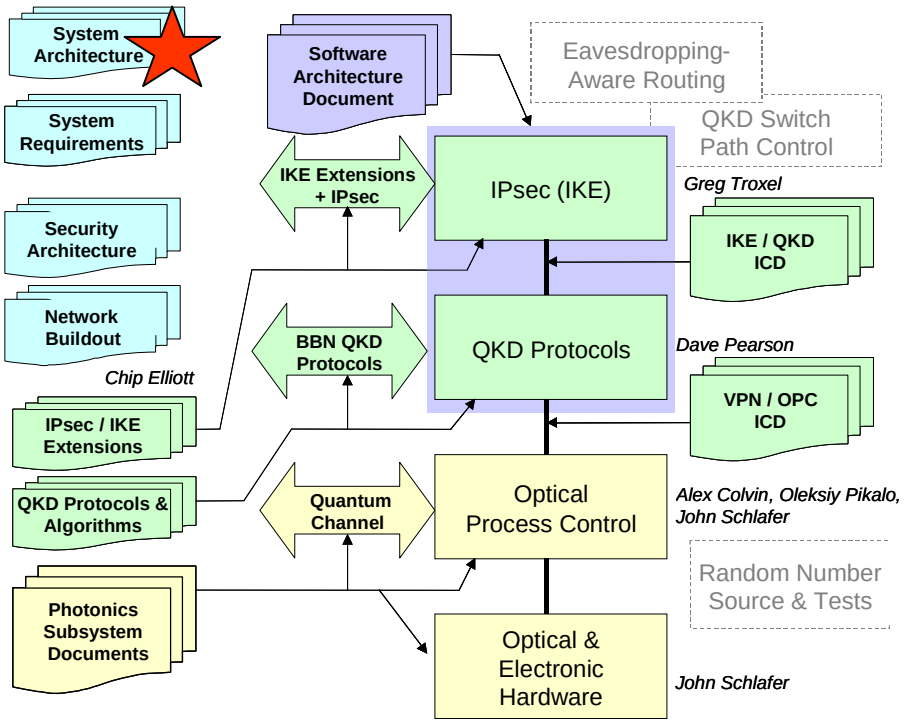


Figure 5-1. This "System Architecture" Document in Context.

5.2 Document Overview

This document is organized into the following sections.

a. Section 1, "Scope," identifies and describes the purpose of this document.

b. Section 2, "Referenced Documents," lists by document number and title all documents referenced by this document.

c. Section 7, "Introduction and Goals," provides an introduction and overview of quantum cryptography, describes what benefits it may bring, and discusses its current limitations. This section also describes our goals for the DARPA Quantum Network, briefly introduces the new types of networks that we intend to create, and outlines the benefits these networks will bring over the prior state of the art.

d. Section 8, “Our Step-by-Step Approach,” defines our year-by-year approach to building the DARPA Quantum Network and specifies what version of the network will be functioning at the end of each Contract Year.

e. Section 9, “Major System Components and their Interactions,” provides a technical overview of the major components within the DARPA Quantum Network, and describes how they fit together and how they interact with the external world. This section is intended as the introductory technical exposition of how the DARPA Quantum Network actually works.

f. Section 10, “The Mark 2 Weak-Coherent Link” describes the Mark 2 Weak-Coherent Link opto-electronic subsystem, along with its Optical Process Control computer and software. It introduces the basic equipment string for the weak-coherent link, explains how weak-coherent quantum cryptography works, describes how optical framing works for the Mark 2 link, and summarizes the basic functions of the Optical Process Computer. This section also describes the interface between the Optical Process Control (OPC) subsystem and the QKD protocol suite running in the Virtual Private Network (VPN) computer.

g. Section 11, “The Mark 1 Entangled Link,” describes the Mark 1 Entangled Link opto-electronic subsystem, along with its Optical Process Control computer and software. It introduces the basic equipment string for the entangled link, explains how entangled quantum cryptography works, describes how optical framing works for the Mark 1 Entangled Link, and summarizes the basic functions of the Optical Process Computer.

h. Section 12, “Random Numbers – Generation and Testing” outlines the critical role of cryptographic-quality random numbers in the DARPA Quantum Network, identifies the sub-components in which random numbers are used, and describes the generation and testing of these random numbers.

i. Section 13, “The BBN QKD Protocols,” presents a technical introduction to the QKD protocols and algorithms implemented in the DARPA Quantum Network. It describes each of the protocols and algorithms in high-level form and shows how these individual sub-components are assembled to form an entire QKD protocol stack. This section also describes the interface between the QKD protocol suite and the IPsec protocol suite.

j. Section 14, “The BBN Key Relay Protocols,” introduces the Key Relay protocols used in the DARPA Quantum Network. These protocols may be used to implement a “trusted network” in which some nodes act as relays for key material between nodes that connect directly share a QKD link, e.g., relays between freespace and fiber-based QKD endpoints.

k. Section 15, “QKD over Optical Switches,” describes a suite of novel protocols and algorithms that enable the “untrusted” version of the DARPA Quantum Network. These protocols and algorithms allow QKD endpoints to set up, monitor, and tear down virtual circuits for QKD photons through a series of one or more passive optical switches.

l. Section 16, “The BBN QKD Routing,” describes novel technology by which nodes in the Quantum Network become aware of a too-high level of noise on the quantum key distribution link, which may indicate the presence of eavesdropping, and how they then “route around” these eavesdropped links.

m. Section 17, “The IPsec Protocol Suite,” provides a basic introduction to the role of IPsec and the Internet Key Exchange (IKE) protocol suite in offering secure communications through an untrusted Internet, and describes our extensions to IKE that marry quantum cryptography into the Internet’s overall security architecture.

n. Section 18, “Analyses of System Performance,” provides a brief synopsis of our analytic modeling of system throughput based on a number of input parameters.

o. Section 0, “Attachment A - Notes and Acronyms
p. ,” provides a list of abbreviations and acronyms, with their definitions, as used in this document. It also contains any additional notes needed for this document.

5.3 Government Rights

The U. S. Government enjoys full government rights to the contents of this document and may freely use any portion of the work with or without attribution. BBN Technologies does retain copyright in this work, however, and the contents may not be used by non-governmental organizations without express written permission from BBN.

Note further that BBN, Boston University, and Harvard University have filed patent applications for certain aspects of the system described in this document, and expect to file additional applications in the foreseeable future. The U. S. Government enjoys full government rights to all such patents that are based on work funded under government contract.

6 Referenced Documents

The following documents of the exact issue shown form a part of this design document to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this document, the contents of this document will be considered superseding.

Government Documents

FIPS 140-2	*Security requirements for Cryptographic Modules*, June 2001. Federal Information Processing Standard, National Institute of Standards and Technology (NIST). http://csrc.nist.gov/publications/fips/

Internet Documents

RFC 1750	Randomness Recommendations for Security
RFC 2401	Security Architecture for the Internet Protocol
RFC 2409	The Internet Key Exchange (IKE)

Other External Documents

None.

Other Documents for the DARPA Quantum Network

This section presents the formal document tree for the DARPA Quantum Network. **Figure 6-1** displays the full set of technical documents along with a schematic indication of the coverage area for each of the documents. Documents are indicated by stacked icons with scalloped bottoms. (Rectangles indicate major system components; arrows indicate major horizontal interfaces.) Items indicated in dashed gray have not yet been incorporated into the system architecture.

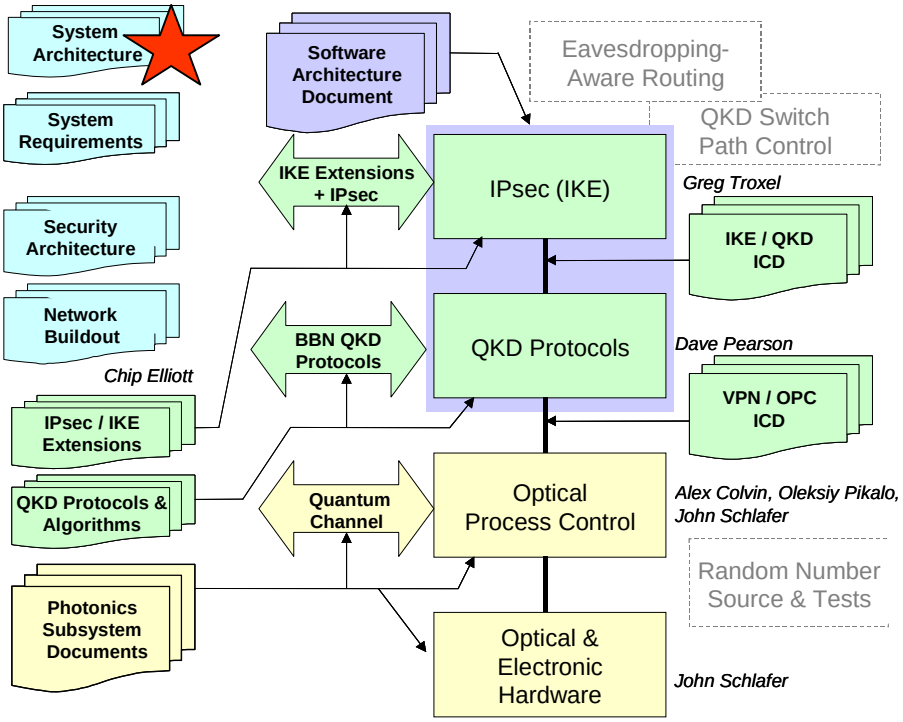


Figure 6-1. Document Coverage for the Quantum Network.

Document Title	Document Contents
System Architecture Description	Overall description of the entire “Quantum Network” system, why it exists and what it does, along with a high-level description of the major sub-systems and their interfaces.
System Requirements Document	List of all requirements on the Quantum Network and its sub-systems, both at the system level and at the level of individual components.
Security Architecture	Closely tied to System Architecture with special focus on security-specific concerns in the overall system. Detailed descriptions of those design issues that are critical for their security implications.
Network Buildout	Documents current status of DARPA Quantum Network buildout, plus historical snapshots of earlier stages of the buildout.
IPsec / IKE Extensions	Detailed descriptions of all Internet protocols and algorithms developed for the Quantum Network, in particular extensions to the IPsec and IKE

	protocol suites.
QKD Protocols and Algorithms	Detailed descriptions of all QKD protocols and algorithms developed for the Quantum Network, including both new techniques and extensions to existing protocols and algorithms.
Photonic Subsystem Documents	Detailed descriptions of each of the photonics subsystems in the Quantum Network, including descriptions of the photonics setup and its associated electronic equipment, the LabView program that controls this equipment, and descriptions of the physical and frame-level encoding of qubits “on the wire.” This document will contain separate volumes for each kind of photonic setup in the Quantum Network, i.e., over time it will grow to include descriptions of the weak-coherent link, the entangled link, the photonic switches, etc.
Software Architecture Document	Descriptions of how the QKD and IPsec/IKE components are implemented, including software diagrams, calling sequences, and descriptions of how the relevant data is stored and manipulated.
IKE / QKD Interface Control Document (ICD)	Formal definition of the interface between the Internet (IKE) and QKD software entities. This document defines exactly what data is transmitted between these entities, and how it is transferred.
VPN / OPC Interface Control Document (ICD)	Formal definition of the interface between the QKD and LabView software entities. This document defines exactly what data is transmitted between these entities, and how it is transferred.

7 Introduction and Goals

This section provides an introduction and overview of quantum cryptography, describes what benefits it may bring, and discusses its current limitations. This section also describes our goals for the DARPA Quantum Network, briefly introduces the new types of networks that we intend to create, and outlines the benefits these networks will bring over the prior state of the art.

Our Technical Concept
We are designing and building the world's first Quantum Network, delivering end-to-end network security via high-speed Quantum Key Distribution, and testing that Network against sophisticated eavesdropping attacks. As an option, we will field this ultra-high-security network into commercial fiber across the metro Boston area and operate it between BU, Harvard, and BBN.

7.1 Why Do Quantum Cryptography?

The information infrastructure of the United States is vital to the prosperity and security of the nation. An exponentially growing flow of private and confidential data transmitted over telecommunication channels has brought with it a heightened concern for its protection. However, significant challenges to the security of that infrastructure are already apparent.

Modern networks generally rely on one of two basic cryptographic techniques to ensure the confidentiality and integrity of traffic carried across their links: symmetric (secret) key and asymmetric (public) key. Indeed today's best systems generally employ *both*, using public key systems for authentication and initial exchange of secret "per session" keys, and then encrypting all or part of a traffic flow with these nonce keys. Certain other systems transport secret keys "out of channel," e.g. via courier, as in classical cryptography.

Unfortunately both techniques are subject to growing concerns. Systems that rely on public key techniques may be vulnerable to adversaries with greatly augmented computing power, e.g., from quantum computers, which can break the public key encryption algorithms. Those that transport secret keys "out of channel" suffer from a different woe, namely that the ever-increasing traffic flows create an ever-increasing burden on the key distribution system, with concomitant growth in both costs and chances for compromise.

The potential vulnerability of public key techniques is particularly troubling. This family includes many well-known algorithms such as RSA, Diffie-Hellman, elliptic curves, etc., all of which rely on the supposed computational infeasibility of undoing a "one way" computation. For instance, classic RSA assumes that it is vastly easier to multiply two large primes to determine their product than it is to factor this product back into its constituent primes. Unfortunately all such arguments collapse if and when quantum computation becomes possible.

This has extremely important practical consequences since most of today's cryptographic systems rest upon this (perhaps invalid) foundation. Indeed, the realization that quantum computation might *someday* be possible immediately casts a shadow over today's cryptographic systems, because an adversary could record encrypted traffic today, and then use quantum computation in the coming years to break it. If there is even the slightest real possibility that this could be feasible, it is extremely important to find new safeguards for sensitive traffic *now*.

7.2 What is Quantum Key Distribution (QKD)?

Fundamental aspects of quantum mechanics – the uncertainty principle, unitarity, and the Einstein-Podolsky-Rosen violation of Bell's inequalities – suggest a new paradigm for secure communications: quantum cryptography. Initial experiments seem to confirm this paradigm. Assuming that the theoretical models continue to be confirmed in the use of actual devices, the fundamental laws of nature can be invoked to assure the confidentiality of transmitted data.

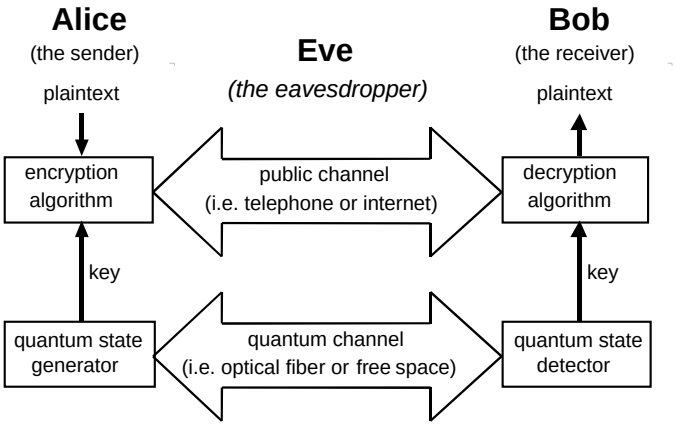


Figure 7-1. Quantum Key Distribution with Alice, Bob, and Eve.

Quantum cryptography provides a new means for distributing secret keys, often termed Quantum Key Distribution (QKD). QKD is showing great promise but it is still in its infancy. The currently implemented techniques are limited to line-of-sight links or relatively short point-to-point fiber connections (48km). Extension of either of these techniques to distant nodes would be expensive and difficult, requiring satellite links or frequent trusted intermediary nodes. Either method is ultimately vulnerable to disruption by an attack against the interlink points.

7.3 Strengths and Drawbacks of Current State-of-the-Art in QKD

In abstract terms, QKD offers a technique for coming to agreement upon a shared random sequence of bits within two distinct devices, with a very low probability that other devices (eavesdroppers) will be able to make successful inferences as to those bits' values. In specific practice, such sequences are then used as secret keys for encoding and decoding messages between the two devices. Viewed in this light, QKD is quite clearly a key distribution technique, and may profitably be compared against other techniques for key distribution such as trusted couriers, algorithmic methods such as Diffie-Hellman, and so forth. More broadly, one can rate QKD's success against a number of important goals for key distribution, as marshalled in the following paragraphs.

Privacy of Keys. QKD offers significant advantages in this regard and indeed this is the main reason for interest in QKD. Public key systems have suffered from an ongoing uncertainty that decryption is mathematically or algorithmically intractable. Classic secret key systems have suffered from rather different problems, namely, insider threats and the logistical burden of distributing keying material. Assuming that QKD techniques are properly embedded into an overall secure system, they can provide automatic distribution of keys that may offer security superior to that of its competitors.

Authentication. When delivering secret keys to someone, it's very important not to give them to the wrong person! QKD is a key agreement primitive that does not in itself provide authentication. Current strategies for authentication in QKD systems include prepositioning of secret keys at the distant device, to be used in hash-based authentication schemes, or hybrid QKD-public key techniques. Neither approach is entirely appealing. Prepositioned secret keys require some means of distributing these keys before QKD itself begins, e.g., by human courier, which of course may be costly and logistically challenging. In addition, this scheme appears open to denial of service attacks in which an adversary forces a QKD system to exhaust its stockpile of key material, at which point it can no longer perform authentication. Hybrid QKD-public key schemes, on the other hand, inherit the possible vulnerabilities of public key systems to cracking via quantum computers or unexpected advances in mathematics.

Sufficiently Rapid Delivery of Keys. Key distribution systems must deliver keys fast enough so that the encryption devices that employ these keys do not run out of keying matter. This is obviously a race between the rate at which keying material reaches the cryptos, and the rate at which the cryptos use up keying material in the course of their encryption or decryption activities. Today's QKD systems achieve on the order of 1,000 bits/second throughput for keying material, at best, and often run at much lower rates. This is unacceptably low if one uses these keys in certain ways, e.g., as one-time pads for high-speed traffic flows. However it may be acceptable if the keying material is used as input for less secure algorithms such as the Advanced Encryption Standard. On the whole, however, it would be beneficial if QKD delivery rates could be increased by at least several orders of magnitude.

Robustness. Since keying material is essential for secure communications, it is extremely important that the flow of keying material not be disrupted, whether by accident or by the deliberate acts of an adversary (i.e. by denial of service). Here QKD has provided a highly fragile service to date since QKD techniques have implicitly been employed along a single point-to-point link. Thus if that single link were disrupted, whether by active eavesdropping or simply by fiber cut, all flow of keying material would cease. We argue that a meshed QKD network is inherently far more robust than any single point-to-point link since it offers multiple paths for key distribution. If one link is disrupted or eavesdropped, the network can automatically route around the disruption.

Distance- and Location-Independence. Ideally, any entity would be able to exchange keying material with any other entity in the world. Rather remarkably, the Internet's security architecture does offer this feature – any computer in the world can form a security association with any other, exchanging keys through the Internet IPsec protocols. This feature is notably lacking in QKD, which requires the two entities to have a direct and unencumbered path for photons between them, and which can only operate for a few tens of kilometers through fiber.

Resistance to Traffic Analysis. Adversaries may be able to perform traffic analysis on a key distribution system in order to understand the relationship between communicating entities. For instance, a heavy flow of keying material between two points might indicate that a large volume of confidential information flows, or will flow, between them. It may be desirable to make such analysis as difficult as possible. Here QKD in general has had a rather weak approach since most setups have assumed dedicated, point-to-point QKD links between communicating entities which has thus clearly laid out a map of the underlying key distribution relationships.

As important guidelines of our overall research agenda, we are working to strengthen QKD's performance in these weaker areas. In some instances, this involves the introduction of newer QKD technologies; for example, we hope to achieve rapid delivery of keys by introducing a new, high-speed source of entangled photons. In other instances, we rely on an improved system architecture to achieve these goals; thus, we tackle distance- and location-independence by introducing a network of trusted relays.

7.4 What is the Quantum Network and Why is it a Good Idea?

The Quantum Network is a key distribution fabric built of network switches and links that are secure against eavesdropping and highly resistant to disruption. Such a key distribution network can reliably and very securely transport keying material between sets of authorized and cooperating network endpoints. To this end, we need QKD-based switching and repeating nodes that can send key material "end to end" between multiple users and yet will be constantly self-monitored for eavesdropping and other events (e.g. fiber cuts) and route around links that exhibit such problems.

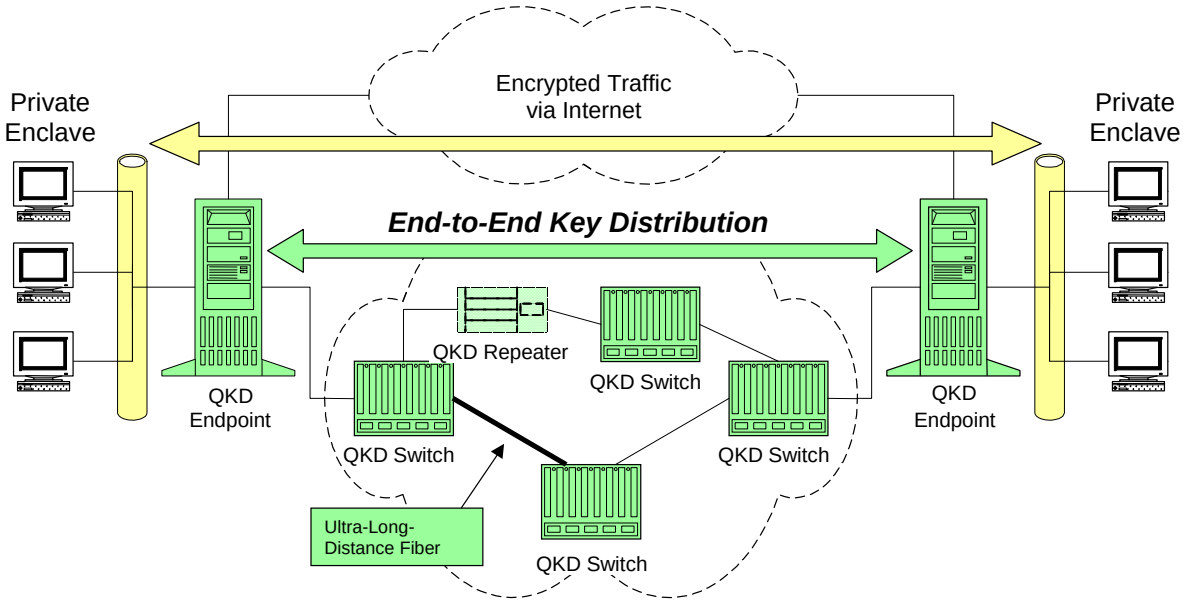


Figure 7-2. Equipment Diagram for the Quantum Network.

Our security model is the Virtual Private Network (VPN). Conventional VPNs use both public-key and symmetric cryptography to achieve confidentiality (privacy) and authentication/integrity (knowing that packets come from an authorized VPN peer and have not been modified in transit). Public-key

mechanisms support key exchange or agreement, and authenticate the endpoints. Symmetric mechanisms (e.g. 3DES, SHA1) provide traffic confidentiality and integrity. Thus VPN systems can provide confidentiality and authentication / integrity without trusting the public network interconnecting the VPN sites. System strength relies on the difficulty of breaking the weaker of the two (public-key key agreement & authentication, symmetric ciphers & hashes).

Perfect Forward Secrecy. With quantum computation, we must assume adversaries can break public-key cryptography and threaten the confidentiality lifetime of our encrypted traffic. Breaking the key agreement primitive (e.g. Diffie-Hellman, DH) ten years hence enables an adversary to decrypt all past traffic. Breaking public-key authentication (e.g. DSA or RSA) is by contrast a lesser threat since it allows impersonating endpoints (and thus stealing traffic) only from that point forward. Thus the more pressing goal is to ensure Perfect Forward Secrecy (Back-Traffic Protection) by new key agreement techniques.

Step-by-Step Approach. We proceed accordingly. First, we replace the DH key agreement primitive with QKD but retain all other mechanisms, including public-key authentication, of the Internet Key Exchange (IKE, the standard IPsec key management protocol). The result will be secure against quantum cryptanalysis of DH, but not failure of DSA or symmetric ciphers. This is an important result: it buys time for further improvements while denying an enemy breaking DH in (say) 2015 all of our traffic before 2015! Then we add symmetric authentication with pre-placed keys and hash functions, and modify IKE to negotiate use of one-time pads based upon QKD for extremely sensitive data, rather than symmetric ciphers with keying material derived from QKD.

End-to-End. Gateways form end-to-end security associations across a public network (Figure 7-2). Unlike conventional systems, our gateways will use two public networks: one of QKD switches for key distribution, and the normal Internet for traffic. There is no fundamental reason why a host itself could not be a QKD endpoint, so security associations would be end-to-end with respect to hosts, not just organizational security gateways.

Security At All Levels. Security issues cut across physics, network security, information assurance, and optical engineering, and must be addressed both on the component level and system-wide. Because QKD is based on quantum physics, QKD techniques must be subjected to rigorous theoretical analysis and eavesdropping experiments. We have organized our research accordingly.

7.5 A Single Point-to-Point QKD Link in its Network Context

Figure 7-3 presents a simplified, block diagram of a point-to-point QKD link as it would likely be employed in practice. Here the QKD link supports secure communications between two private enclaves so that they may exchange confidential information through a public communications network such as the Internet or the global telephone system. Each enclave is typically a collection of one or more local Ethernets; the whole diagram thus represents a widely deployed type of secure networking, e.g., one that securely links a branch office to a corporate headquarters.

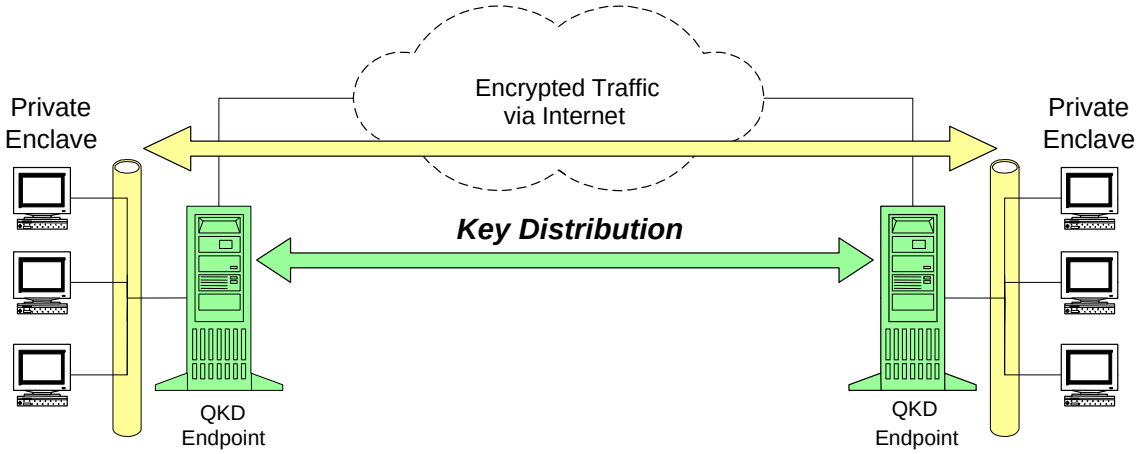


Figure 7-3. Simplified Block Diagram of a Point-to-Point QKD Link in Context.

Such secured communications are often implemented via specialized devices such as cryptographic gateways so that one need administer only a single device in order to establish or monitor external security for a given private enclave. These gateways are responsible for setting up security associations (and thus encrypted tunnels) with authorized distant gateway(s), for encrypting all local traffic before it is injected into the public network, and for decrypting traffic received from the public network before sending it onwards, in the clear, within the destination enclave. In general, such systems require two distinct communications paths: one for the cryptographic keys themselves, the other for the encrypted message traffic. In conventional technology, keys may be distributed via an out-of-band channel such as couriers, or via an in-band channel using techniques such as Diffie-Hellman key exchange. In the Quantum Network, keys are distributed via a smorgasbord of out-of-band QKD techniques (e.g. weak-coherent and entangled), and these cryptographic gateways thus act as QKD endpoints because they contain one or more QKD devices apiece.

Today's Internet offers a well-defined security architecture, IPsec, that defines the protocols, algorithms, databases, and policies required for secure communication. IPsec provides all the tools needed for secure communication between cryptographic gateways or indeed between individual computers on the Internet. Among other things, IPsec defines how two endpoints authenticate each other, exchange keys, and encrypt and decrypt messages flowing between these endpoints. Thus in order to secure Internet traffic via quantum cryptography, one must marry QKD technology into the overarching, and already well-established, Internet security architecture.

Figure 7-4 resolves this basic setup into considerably more detail and represents the first stage in our Quantum Network research program. The basic concepts are not difficult: a) two QKD endpoints establish communications via a dedicated fiber or wavelength for the quantum path, and via the Internet for messaging; b) the transmit side prepares and transmits raw keys, from which both sides come to agreement on a shared, secret key; c) this secret key is then employed in the VPN cryptographic gateway for protecting message traffic that will transit the Internet within secured IPsec tunnels.

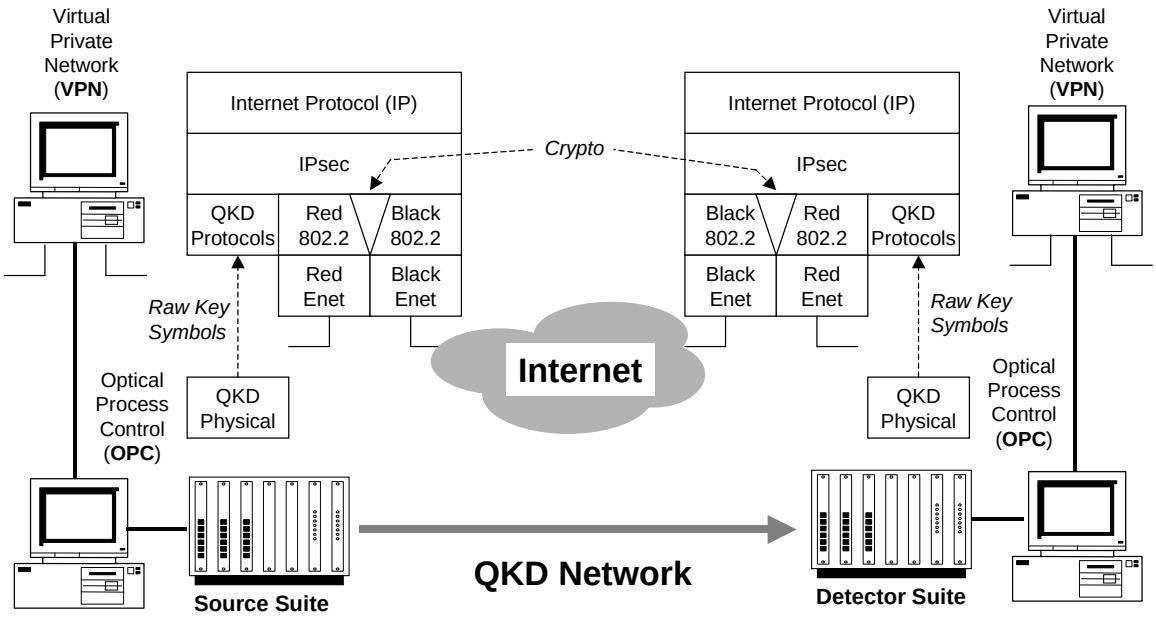


Figure 7-4. System Architecture for a Point-to-Point QKD Link in Context.

As in the simpler schematic, this diagram depicts two private enclaves – at the left and right edges of the drawing – connected via two distinct communications networks, namely, the Internet for message traffic, and a fiber optic QKD channel for key distribution. Each VPN cryptographic gateway is implemented via a pair of specialized computers. The Virtual Private Network (VPN) computer implements the non-realtime portions of the gateway, and in particular both the QKD protocols and the Internet Protocol suite. It also contains the encryption device for message traffic. The Optical Process Control (OPC) computer runs LabView programs that perform management of the optical and electronic devices that comprise the QKD laser source suite and APD detector suite. These two computers are linked via a private Ethernet that carries control traffic between them, including the raw key symbols as transmitted or received via the QKD link.

This system can be most easily understood by following its operation during routine practice. At such time, the VPN and OPC computers operate in only loose synchrony. In essence, the OPC source and detectors suites provide a continuous stream of raw key symbols to their corresponding VPN computers. These symbols include both the basis used for a given bit (as transmitted or received) and the bit's value, along with QKD framing information. As the VPN computers receive these frames of raw key symbols, they perform a suite of well-known QKD protocols (sifting, error correction, privacy amplification, etc.) to derive the actual "good" key bits. These good bits continuously accumulate in a reservoir of shared keying material within each QKD endpoint. Referring back to Figure 7-4, this keying material is then available for use via the IPsec protocol suite where it can be used to encrypt one or more secure tunnels through the Internet. In particular, the keying material is fed to an encryption device (crypto) for use in encrypting or decrypting IP datagrams as they pass through the VPN computer. For traffic flows that must be very highly secured, a one-time pad (Vernam cipher) approach may be employed. For less critical traffic flows, the key may be used as input to an encryption algorithm such as the Advanced Encryption Standard (AES), Triple Data Encryption Standard (3DES), etc. As has been discussed by others, the QKD

mechanisms may thus be employed to rekey such algorithms at fairly high rates, e.g., once per second, which may improve the overall system security.

In short, the QKD keying material is employed in both VPN computers as keys for the local crypto device. Then as traffic flows enter the VPN computer in the clear from the private enclave (conventionally known as the "red" side of the gateway), they pass through this crypto and become encrypted. These encrypted datagrams are then carried through the "black" public Internet, or any other suitable communications network, until they are received at the distant VPN gateway, where they are passed through another crypto and thus decrypted, and then sent once again in the clear onto the "red" private enclave at the distant location.

Figure 7-5 provides a finer-grained schematic of the multiple layers of protocols and algorithms in a useful QKD protocol suite. There are a number of options for most of these sub-layers. For example, encoding options include BB84 vs. B92, polarized vs. interferometric encodings, weak-coherent vs. entangled, round-trip vs. one-way, etc.; error detection and correction options include Cascade (BS92), simple parity as in Los Alamos, etc. Most of these options have been previously implemented, either in functioning QKD systems or in simulations, with the apparent exception of authentication. We intend to implement a full suite of these protocols and algorithms, in either C or C++ for performance reasons, with fully documented "mix and match" interfaces so that QKD protocol stacks can be assembled as desired, and to make these implementations freely available. This implementation is planned to include continuous authentication via an initial seed of key material at the endpoints, periodically refreshed by drawing off unused keying material that has been disseminated via QKD links in operation. We also intend to create new protocols and algorithms in this area, with an eye to improved efficiency.

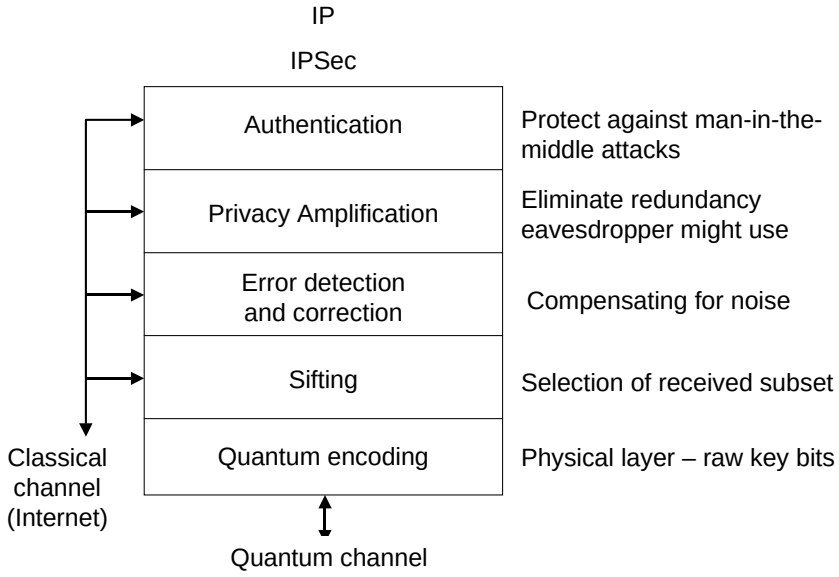


Figure 7-5. Internal Structure and Functionality of QKD Protocol Suite.

7.6 A Full "Trusted" QKD Network

As useful as a QKD point-to-point link may be, it still suffers from striking drawbacks. First, it is geographically constrained by the distance over which a single link may be operated. Fiber attenuation limits terrestrial links to 50 km or less in practical applications. Free-space links, e.g. to airborne relays or satellites, may allow wide-area or even transcontinental links but still do not permit truly global coverage. Second, isolated point-to-point links are subject to simple denial-of-service attacks such as active eavesdropping or cutting the fiber. Third, in practice it may be prohibitively expensive to establish pairwise, dedicated point-to-point links between all private enclaves that wish to communicate with each other.

To a surprisingly large extent, these drawbacks can be mitigated by organizing a number of QKD links into a QKD network, which is the second major step in our research program. Figure 7-6 depicts a QKD network in highly schematic form. As in previous diagrams, QKD endpoints can be seen to the left and right edges of the diagram. Behind these cryptographic endpoints lie private enclaves. In contrast with the point-to-point links described before, however, the QKD endpoints are now linked via a mesh of QKD relays or routers.

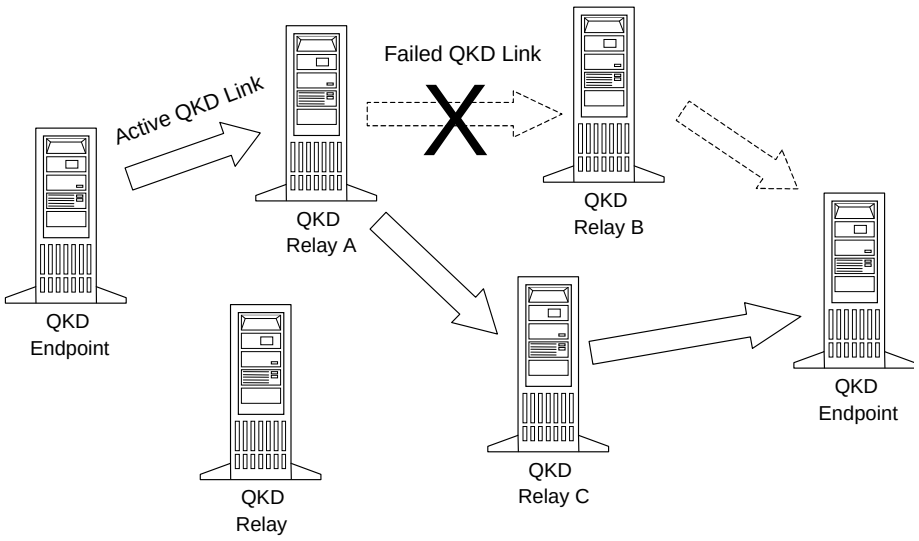


Figure 7-6. QKD Network with Trusted Relays.

As depicted, this form of a QKD network is composed from a collection of point-to-point QKD links. Thus the leftmost QKD endpoint exchanges keying material with Relay A, which in turn exchanges keys with Relays B and C, etc. When a given point-to-point QKD link within the relay mesh fails – e.g. via a fiber cut or too high a level of eavesdropping or noise – that link is abandoned and another used instead. Thus the overall QKD network can be engineered to be resilient even in the face of active eavesdropping or other forms of denial-of-service attacks.

Such QKD networks can be built in several ways. In one variant, the QKD relays may transport only keying material but never message traffic. Thus after the various relays have established pairwise agreed-to keys along an end-to-end point, e.g., between the two QKD endpoints, they may employ these key pairs to securely transport a key from one endpoint to the other. Such a design may be termed a "key

transport network." In another variant, the QKD relays may transport both keying material and message traffic. Figure 7-7 illustrates this second variant, in which the relays are acting as Internet routers with pairwise QKD mechanisms providing link encryption between the routers. In essence, each IP datagram of message traffic is encrypted once as it transits from the QKD endpoint to its first relay. Then it is decrypted, held in the clear in the relay's memory, and then re-encrypted with a second set of keys and sent onwards to the next relay. This operation proceeds, hop by hop, until the datagram is finally received at the destination endpoint and sent onwards to the attached private enclave.

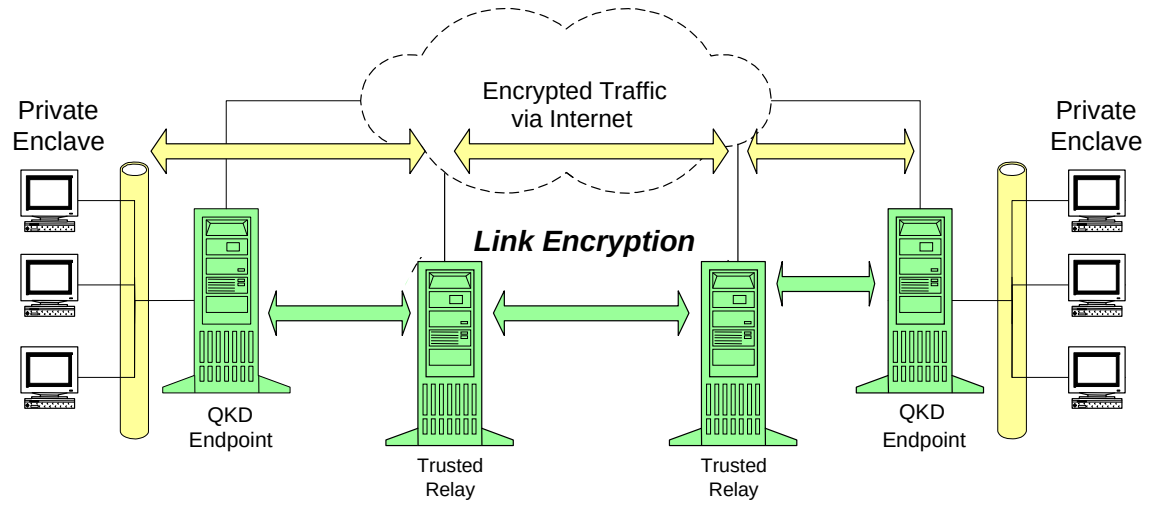


Figure 7-7. QKD Network with Trusted Relays and Link Encryption.

Such QKD networks bring important benefits that greatly mitigate the drawbacks of point-to-point links enumerated at the start of this section. First, they can greatly extend the geographic reach of a communications network secured by quantum cryptography, since wide-area networks can be created by a series of point-to-point links bridged by active relays. These links can further be heterogeneous transmission media, i.e., some may be through fiber while others are freespace. Thus in theory such a network could provide fully global coverage. Second, they lessen the chance that an adversary could disable the key distribution process, whether by active eavesdropping or simply by cutting a fiber. A QKD network can be engineered with as much redundancy as desired simply by adding more links and relays to the mesh. Third, QKD networks can greatly reduce the cost of large-scale interconnectivity of private enclaves by reducing the required (N x N-1) / 2 point-to-point links to as few as N links in the case of a simple star topology for the key distribution network.

Such QKD networks are by no means panaceas, however. Their prime weakness is that the relays must be trusted. That is, since keying material and – directly or indirectly – message traffic are available in the clear in the relays' memories, these relays must be prevented from falling into an adversary's hands. In practice, they would need to be in physically secured locations and guarded if the traffic were truly important. A related, but perhaps more subtle, drawback is that all users in the system must trust the network (and the network's operators) with all keys to their message traffic. Thus even if a pair of users have unusually sensitive traffic, they must expand the circle of those who can be privy to it to include all machines, and probably all operators, of this QKD network that is used to transport this sensitive traffic.

This is obviously undesirable for truly sensitive traffic, where the circle of trust should be kept as small as possible.

7.7 An "Untrusted" QKD Network with Photonic Switches

As in classical cryptography, an end-to-end approach is likely to provide the most satisfactory architecture for disentangling the users' keying material for secured traffic flows from the network that transports such flows. The third step in our research program thus introduces unamplified photonic switches into the QKD network architecture in order to provide end-to-end key distribution via a novel mesh of untrusted switches.

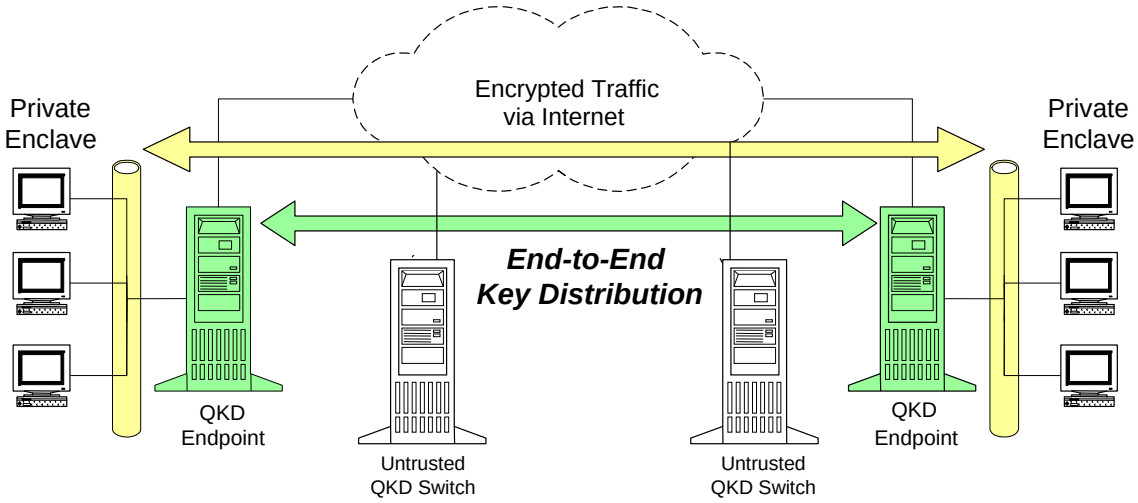


Figure 7-8. QKD Network with Untrusted Photonic Switches.

Figure 7-8 depicts this architecture in schematic form. By contrast with the trusted network architecture, the untrusted QKD switches do not participate in the QKD protocols at all. Instead they are merely used to set up all-optical paths through the network mesh of fibers, switches, and endpoints. Thus in essence a photon from the leftmost QKD endpoint proceeds, without measurement, from switch to switch across the optical QKD network until it reaches the rightmost QKD endpoint at which point it is detected. We currently anticipate that the QKD switches will be built from MEMS mirror arrays, or equivalents, together with novel distributed protocols and algorithms that allow end-to-end path setup across the network, and that (as in untrusted networks) provide a robust means for routing around eavesdropping or failed links.

Untrusted QKD networks have different strengths and weaknesses than trusted QKD networks. Their main strength is that they provide truly end-to-end key distribution; QKD endpoints need not share any secrets with the key distribution network or its operators. This feature may be extremely important for highly secure networks. Their weaknesses appear significant, however. Unlike trusted relays, the untrusted switches cannot extend the geographic reach of a QKD network. In fact, they may significantly reduce the network's reach since each switch adds at least several dB loss to the photonic path. In

addition, it will likely prove difficult in practice to employ a variety of transmission media within an untrusted network, since a single frequency may not work well along a composite path that includes both fiber and freespace links. Untrusted networks may also introduce new vulnerabilities to traffic analysis.

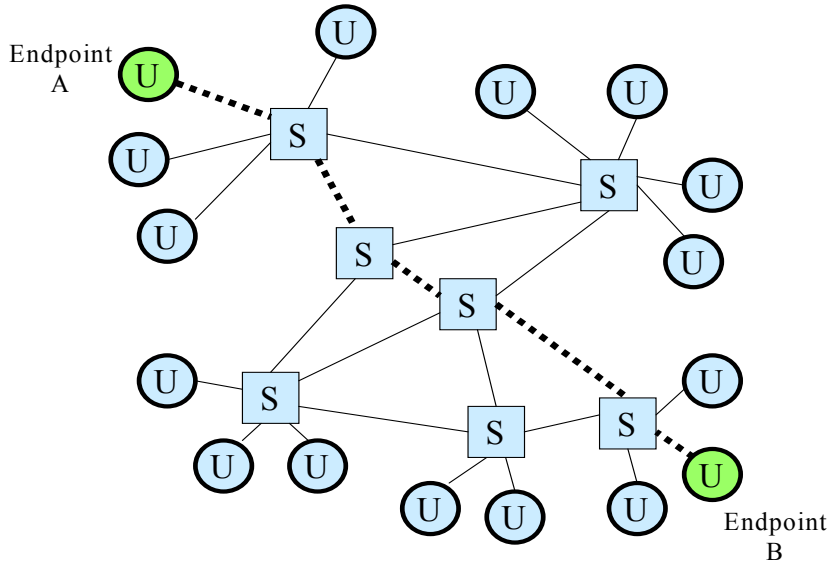


Figure 7-9. End-to-End Security.

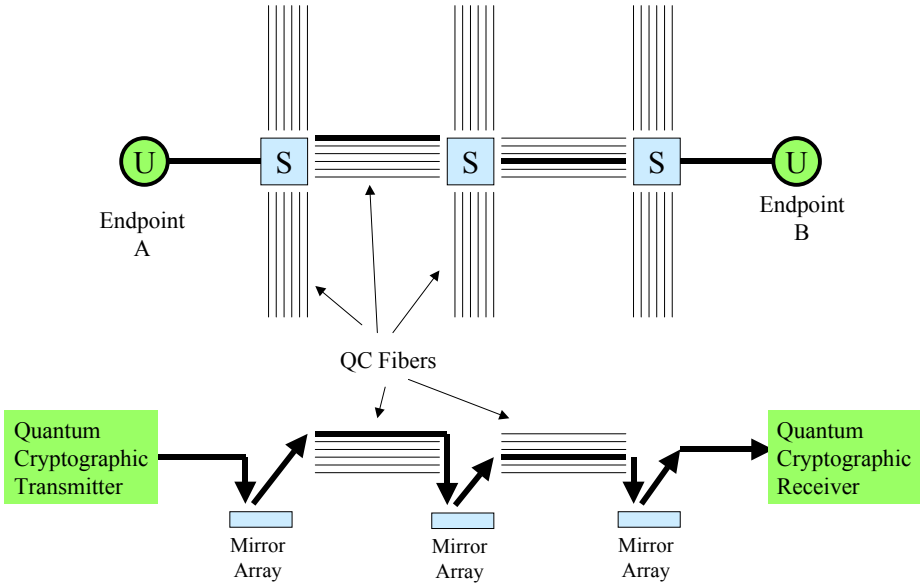


Figure 7-10. End-to-End Security Implemented via Mirror Switches.

7.8 How the Quantum Network Improves QKD

The following table highlights how the DARPA Quantum Network will improve the current state of the art in quantum cryptography, as the project develops over its first three contract years.

Goals for a Key Distribution System	QKD Current Art	Year 1 Single QKD Link (Weak Coherent)	Year 2 Trusted Network & Untrusted Network	Year 3 Network (Entangled Source)
Protection of Keys	✓ Good. QKD appears to provide good protection.	✓ Good. We implement full suite of QKD mechanisms.	✓ Better. Red team activities allow better understand of threats and defenses.	✓ Better. More red teaming, better grasp of physics, better defenses.
Authentication	☐ Poor. Theoretical approaches, no known analysis of overall system effects	✓ Good. Mix of QKD techniques (secret key) and IPsec primitives gives good authentication.	✓ Good.	✓ Good.
Distance- and Location-Independence	☐ Poor. Single link, through fiber or air.	☐ Poor. Single link, through fiber or air.	✓ OK. Trusted network provides relays to extend geographic reach.	✓ OK.
Robustness	☐ Poor. Eaves-dropping or other denial-of-service attacks are easy.	☐ Poor. Eaves-dropping or other denial-of-service attacks are easy.	✓ Good. Multiple links allow routing around eaves-dropping, noise, or faulty links.	✓ Good. Novel switched key distribution protocols allow routing around failures.
Rapid Delivery of Keys	☐ Fair. Rates limited to less than 1,000 key bits / second in practice.	☐ Fair. We implement weak coherent QKD system; speeds similar to prior research systems.	☐ Fair. We implement weak coherent QKD system; speeds similar to prior research systems.	✓ Good? May incorporate better detectors to dramatically improved key delivery rate.
End to End Security	☐ Poor. Only security provided is link encryption.	☐ OK. QKD techniques used for per-session, per-flow traffic keys.	✓ Good. End to end QKD security provided by novel protocols and algorithms, with new QKD Switches.	✓ Good. End to end QKD security provided by novel protocols and algorithms, with new QKD Switches.
Resistance to Traffic Analysis	☐ Poor.	☐ Poor.	☐ Poor.	☐ Poor? This aspect of untrusted network is not yet understood.

8 Our Step-by-Step Approach

This section defines our year-by-year approach to building the DARPA Quantum Network and specifies what version of the network will be functioning at the end of each Contract Year.

8.1 High-Level Overview of Our Approach

Figure 8-1 displays our basic approach in broad, conceptual form. The left edge of the figure represents the start of the project; the right edge represents its conclusion at the end of Contract Year 5. As can be seen, we pursue several tracks in parallel throughout the project, with results of some of these tracks feeding in to the main engineering effort of building the Quantum Network.

In particular, one ongoing effort – depicted as "Quantum Physics and Security Analyses" – will be to better understand the quantum physics that underlies practical quantum cryptography and, as a result, to better understand the exact scope of its security guarantees. This effort is blended with our overall security analysis effort, which employs an informal "red team" and "blue team" effort to continuously devise new attacks on the network as it is being built, and to constantly improve the network security in response to these attacks.

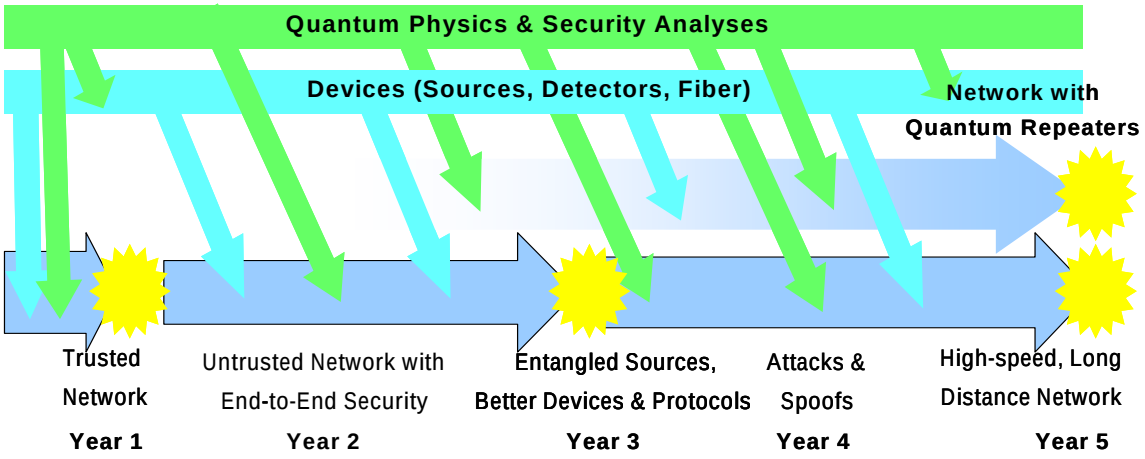


Figure 8-1. High-Level Schematic of our Step-by-Step Approach.

Another ongoing effort – depicted as "Devices" – will be to continuously develop and/or acquire better individual devices. The most important part of this effort is Boston University's development of improved sources based on pairs of entangled photons produced by femtosecond lasers. However we also expect to acquire new detectors and optical fiber from other DARPA QuIST researchers, or indeed other research teams around the world. It may be that we also develop some of these devices ourselves. Whatever the origin of these new devices, after an initial trial and shakedown period, they will be integrated into the overall DARPA Quantum Network.

The main course of our work is depicted as a broad arrow along the bottom edge of the figure. This is the gradual build-up of the full DARPA Quantum Network. Our basic approach here is to get an initial system up and running as soon as possible, and then steadily improve it over the course of the project. We

produce a new release of the DARPA Quantum Network at the end of every contract year as our major deliverable for that year. As shown, we will have a weak-coherent QKD link running in our lab at the end of Year 1, introduce the QKD Switches and QKD key relay to form the first "trusted" and "untrusted" quantum networks in Year 2, and enhance it with an additional link based on Boston University's entangled source in Year 3. As currently envisioned, years 4 and 5 will be dedicated to concentrated attacks and spoofs, together with the defenses against these attacks, and to general improvement of the DARPA Quantum Network by introduction of new types of optical fibers and quantum devices (including perhaps quantum repeaters) as they become available.

8.2 Year 1 – One Weak-Coherent Link Plus Two Simulated Links

In Year 1, we will build the BBN-based lab and perform the first integration of QKD endpoints. These endpoints will employ weak coherent sources and detectors similar to those already demonstrated, and will tie these devices and their associated BB84 protocols into our Linux PC hardware and operating system. We will employ these endpoints in an interim role as "trusted switches," i.e., full IP routers that employ QKD mechanisms for link encryption but which require physical protection of the switch devices in order to guarantee security. As a result, we will have a full Weak-Coherent QKD link operational in the DARPA Quantum Network by the end of Year 1.

Meanwhile, in a parallel effort at BU, we will design and demonstrate high-fidelity entangled sources as stand-alone components. In addition, all teammates will collaboratively draw up detailed plans for the first-generation Quantum Network with high-speed sources and untrusted switches.

Figure 8-2 depicts the state of the DARPA Quantum Network at the end of Year 1. As shown, there will be three QKD Endpoints in the network (Alice, Bob, Charlie). Alice and Bob will be linked by a real, operational Weak Coherent QKD link. The other two QKD links in this version of the network will be simulated (i.e. the Alice-Charlie and Bob-Charlie links). That is, at the higher levels of the protocol stack these links will be fully functional; however the lowest-level transport of qubits will be performed by some means other than an actual single-photon optical channel.

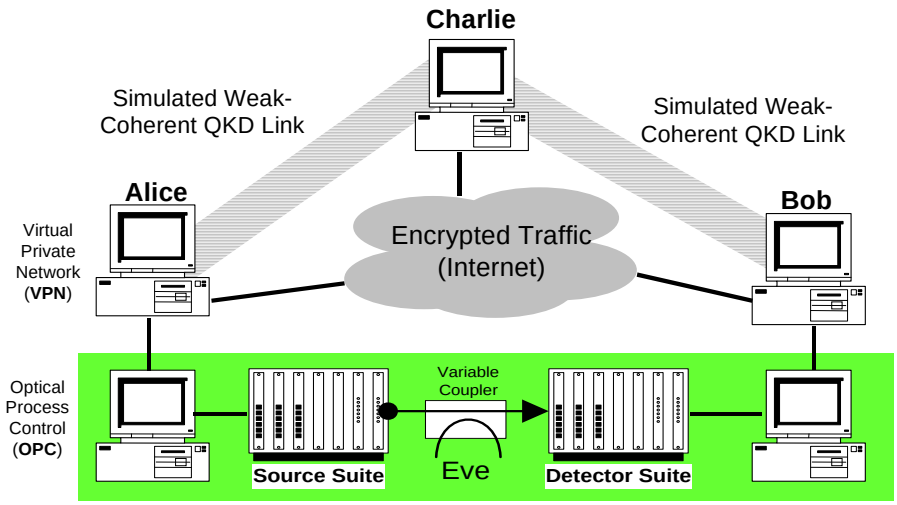


Figure 8-2. Quantum Network Testbed at end of Contract Year 1.

Due primarily to funding constraints, we expect that our Eve station will have only limited physical apparatus at the end of Year 1. Of course it is desirable that Eve have all available technology at its disposal, including but not limited to better versions of the QKD endpoints than are possessed by either Alice or Bob. However this will not be feasible under the first year's budget. Hence we expect to "loan" Eve equipment that has been taken away from Alice or Bob in order to perform our eavesdropping experiments in the first year. In subsequent years, we will gradually build up Eve's equipment suite, but it is likely that we will always need to borrow the newer transmit and receive devices for Eve's use since they may well be expensive enough so that we cannot afford to dedicate a copy of such devices at Eve's station.

Figure 8-3 presents a different sort of diagram for the Quantum Network laboratory at the end of Year 1. This figure portrays the network-level "wiring diagram" for the laboratory and indicates which computers go where and how they are connected. It is intended to emphasize the connectivity that will be exercised during eavesdropping experiments. As shown, the VPN computers for Alice, Bob, Charlie, and Eve are all connected via the lab's internal Ethernet. Eve is able to see clear-text versions of all messages between Alice and Bob, and has complete control of the optical channel (QKD fiber) as it passes from Alice to Bob.

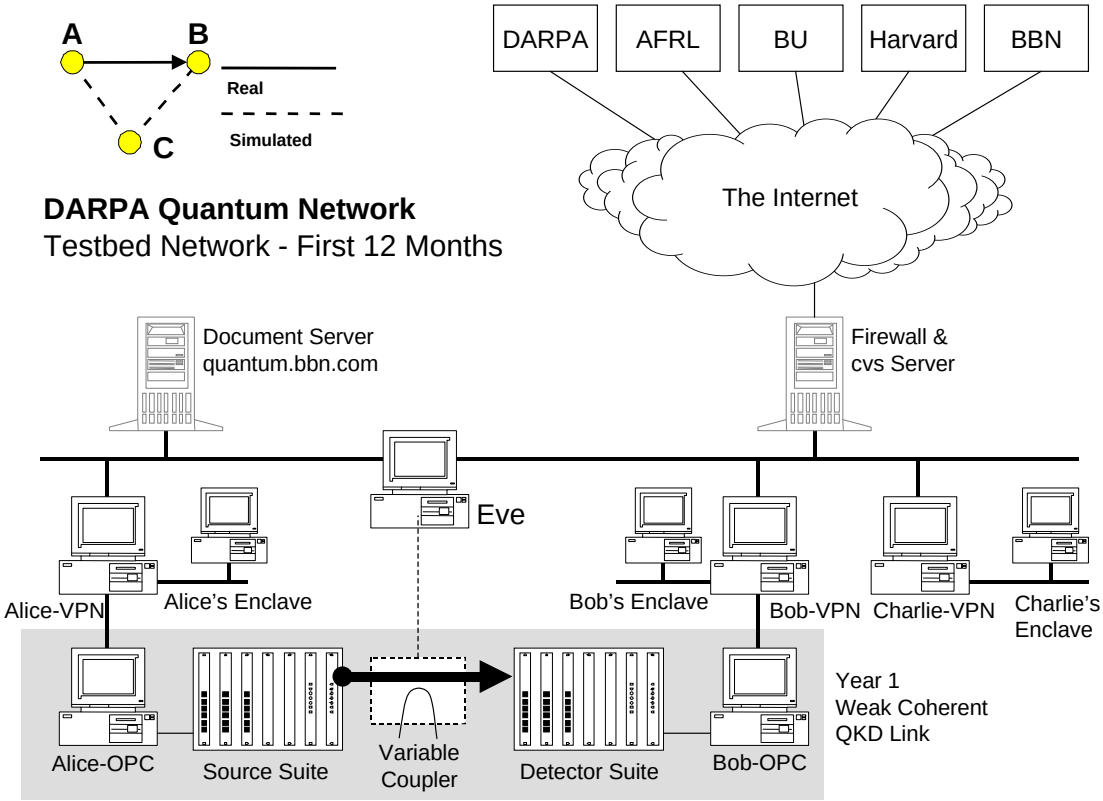


Figure 8-3. Quantum Network Wiring Diagram at end of Contract Year 1.

All three QKD Endpoints are connected to the lab's Ethernet, but only Alice and Bob have a physical QKD link between them. All simulated "single photon" interactions of Alice or Bob with Charlie, therefore, are transported in IP datagrams across the lab's Ethernet rather than via a real QKD link.

Note that the entire Quantum Network laboratory is outside BBN's corporate firewall. We have set it up this way to make it as easy as possible for visiting researchers to come to the laboratory and simply plug in their computers. (BBN corporate policy does not encourage non-employees to readily plug in to BBN's internal corporate network.) We have positioned our own firewall between the laboratory Ethernet and the outside, untamed Internet in order to help protect laboratory computers (and visitors' computers) from the constant barrage of attacks present in the public Internet.

These diagrams also call out a few other computers used in the Quantum Network project. In particular, they show the positions of the firewall that protects the laboratory Ethernet from the untamed Internet to which it's connected; of the cvs repository that maintains controlled versions of all the software and documents used in the project; of the 'quantum.bbn.com' document server that provides ready access to project documents both within the team and to the outside world; and the various computers used by developers.

8.3 Year 2 – Introducing End-to-End Security with Photonic Switching

In Year 2, we will build and demonstrate the first version of our untrusted Quantum Network. This version will contain QKD endpoints and QKD switches, and employ the first generation of protocols that are needed for such a novel network (e.g. QKD circuit setup and eavesdropping-aware routing).

It will employ the Weak Coherent QKD link developed during Year 1. During this year, Boston University will build and demonstrate the 1st generation entangled source, i.e., a mini femto-second source of entangled photons. This will provide an entangled source packaged in a form suitable for realistic deployment.

Figure 8-4 presents a highly notional view of the DARPA Quantum Network at the end of Year 2. We have depicted the MEMS 4 x 4 passive optical switch from OMM Inc. mainly as an existence proof, since we would rather not build a mirror device ourselves. The detailed plans for our Year 2 QKD Switch will be developed during Years 1 and 2, and we will substitute more concrete technical information about the untrusted DARPA Quantum Network as its design becomes clearer.

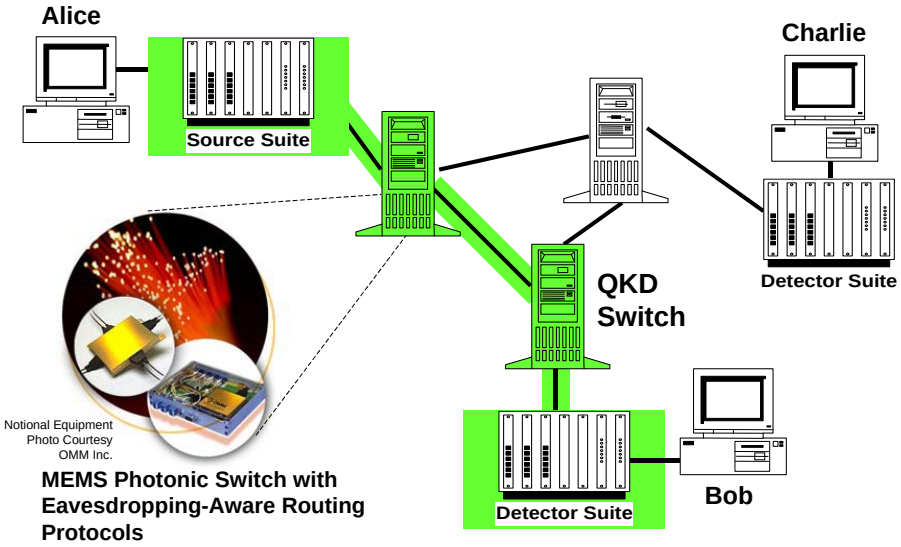


Figure 8-4. Quantum Network Testbed at end of Contract Year 2.

8.4 Year 3 – Adding a Link that implements Entanglement-Based QKD

In Year 3, we will upgrade the existing Quantum Network to add BU's entangled source and the first generation of our new high-speed, authenticated security protocols. The resulting network will thus still consist of trusted switches and untrusted switches, and will incorporate devices and protocols that we expect to employ in the rollout of the Quantum Network. It will now also consist of two real links (the prior Weak Coherent link plus the new entangled link), leaving only one simulated QKD link in the network.

In this year, we will also perform an initial round of eavesdropping experiments and report the results. At Boston University, we will build and demonstrate a simple electro-mechanical mirror device that will form the core of the QKD Switch. In addition, all teammates will perform research that will lead to higher speed sources and better understanding of overall system security.

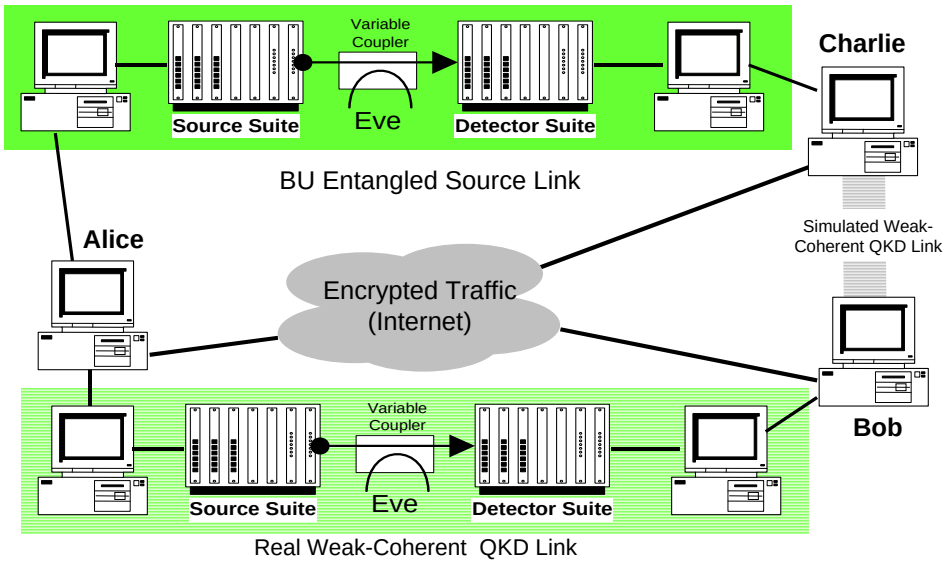


Figure 8-5. Quantum Network Testbed at end of Contract Year 2.

8.5 Year 4 – Concentrated Attacks, Spoofs, and Quantum Hacking

In Year 4, we concentrate our energies on "quantum hacking," on a barrage of attacks and spoofs. We will employ the red and blue teams that have been active since Year 1, and try hard to eavesdrop upon the QKD links without being detected, and/or introduce new signals into these links. At BBN we will revise our protocols in response to what we've learned from these experiments and to general experience with the protocol suite, thus giving rise to the 2nd generation of novel protocols for the Quantum Network. At BU we will investigate, and if possible demonstrate, the very high risk quantum technologies: quantum teleportation, entanglement purification, and Quantum Wavelength Division Multiplexing (QWDM). If any of these techniques appear practicable, we will build lab versions and plan how to integrate them into the Quantum Network. In this year we will also purchase ultra-low loss fiber from Dr. Yoel Fink's team at MIT, or elsewhere, and integrate it into our Quantum Network.

8.6 Year 5 – Ultimate Version of DARPA Quantum Network

In Year 5, we will demonstrate the final versions of our Quantum Network protocols and hardware, and prepare a full report on all aspects of its security (based on both theoretical work and on our concentrated eavesdropping experiments). If feasible, we will introduce QWDM elements into the Quantum Network along with a prototype quantum repeater. These last items should be viewed as stretch goals, however, and may well not be achieved.

9 Major System Components and their Interactions

This section provides a technical overview of the major components within the DARPA Quantum Network, and describes how they fit together and how they interact with the external world. This section is intended as the introductory technical exposition of how the DARPA Quantum Network actually works.

9.1 The Big Picture

Figure 9-1 presents an overall diagram for a single end-to-end link in the Year 3, full-blown DARPA Quantum Network. Green-shaded components will be built specially for the Quantum Network.

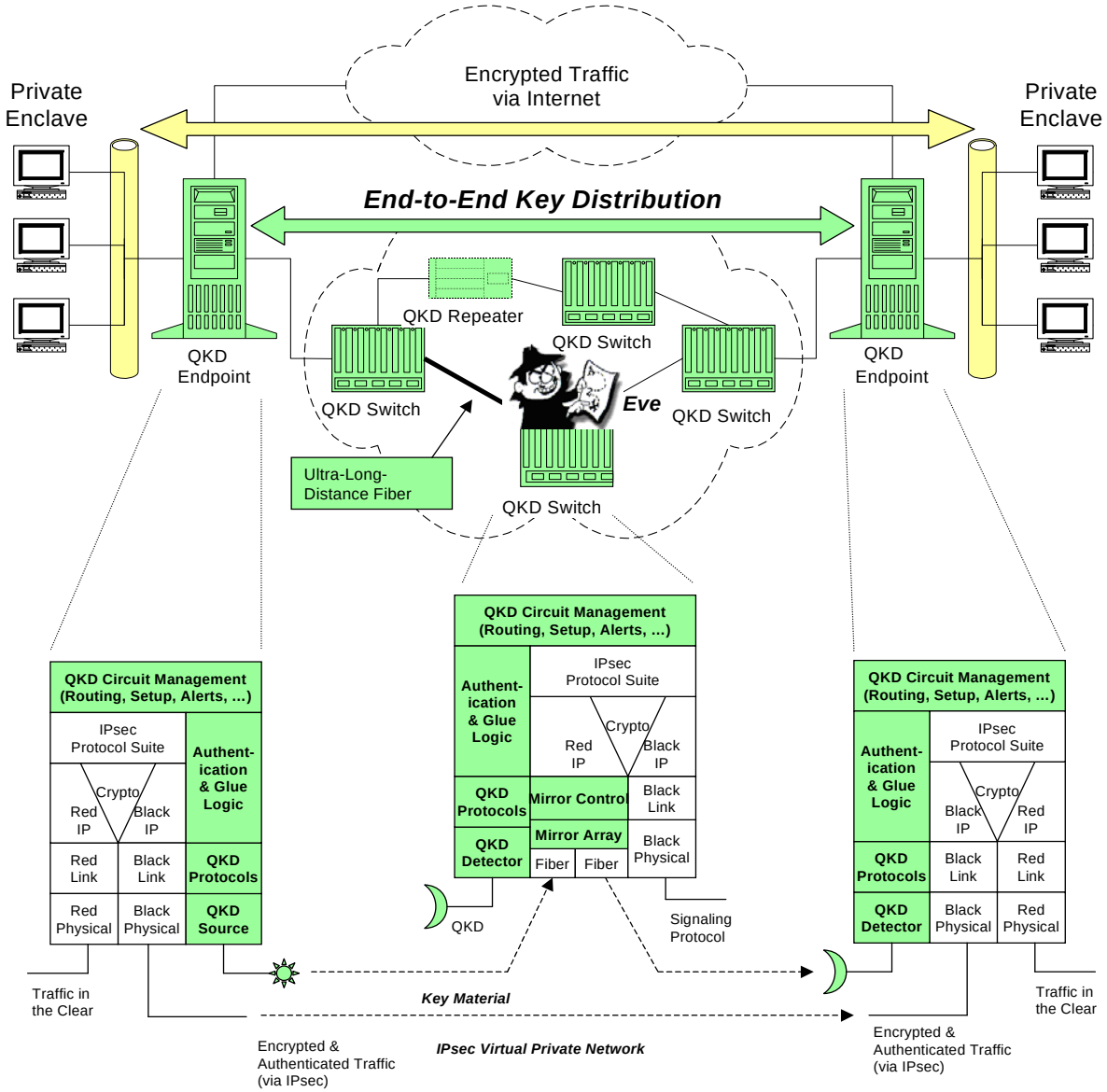


Figure 9-1. Big Picture of (Year 3) System Components in Context.

The top portion of the diagram depicts all the types of equipment (boxes) in the DARPA Quantum Network. Note that the QKD Endpoint equipment contains many of the interesting sub-components such as the weak-coherent source and detector suites, BU's entangled source, and so forth. It also contains much of the new software that is being developed in this project to implement the QKD protocols and algorithms, as well as extensions to the Internet Protocol suite to provide secure communications based on quantum cryptography. Some of this detail is broken out at the bottom portion of the diagram, which presents a set of protocol stack schematics as are generally employed by network designers.

The following table briefly describes those major components that appear in Figure 9-1. It also points out where the equipment will come from (if not developed by the team members) and the contract in which this component will be introduced into the DARPA Quantum Network.

Object	Year	Source	Brief Description
QKD Endpoint	1	Team	A Virtual Private Network (VPN) gateway that incorporates one or more Quantum Key Distribution (QKD) devices such as weak-coherent or entangled sources, or single-photon detectors, along with all necessary protocols and algorithms to support the IPsec protocol suite augmented by quantum key distribution.
Eve	1	Team	An eavesdropping station that is fully capable of intercepting and/or fabricating single photons for the quantum channel, as well as cleartext versions of all messages exchanged among QKD Endpoints and QKD Switches in the DARPA Quantum Network. Eve will begin with relatively simple experimental apparatus but will be continuously augmented as the project unfolds. We will provide a simulated single-photon channel in addition to the actual QKD links so that Eve may (via this simulator) enjoy the nearly unlimited powers that are conventionally granted to her in QKD security research; such powers are extremely difficult to construct in actual hardware!
QKD Source (Weak Coherent)	1	Team	An opto-electronic suite that prepares and transmits frames of single photons via demonstrated weak-coherent techniques, i.e., by employing a highly attenuated telecommunications laser. Our main approach here will be phase-encoded BB84 running at a wavelength suitable for installed telco fibers (1550 nm).
QKD Detector	1	Team	An opto-electronic suite that detects frames of single photons via a pair of gated and thermo-electrically cooled Avalanche Photo Diodes (APDs). Our main approach here will be phase-encoded BB84 running at a wavelength suitable for installed telco fibers (1550 nm) to match the weak-coherent QKD source.
QKD Protocols	1	Team	A full implementation of the documented QKD protocols and algorithms, including BB84 and B92, Cascade (BS92) and simpler parity checks for error correction, and privacy amplification. These protocols will be supplied in a form that allows efficient protocol interactions across the shared Internet communications medium and that allows easy assembly of stacks as desired from the individual protocols or algorithms in the "toolkit."

IPsec (IKE)	1	Team	A version of the Internet Protocol security suite (IPsec), and in particular of the Internet Key Exchange (IKE) mechanisms, that employ the secret key bits provided by the QKD Protocols in the more general context of securing communications via IPsec techniques. In particular, the IPsec suite will be modified to support Virtual Private Networks (VPNs) protected by QKD-supplied secret key material.
QKD Source (Entangled)	2	Team	A novel opto-electronic suite that prepares and transmits frames of single photons from a source of entangled photons prepared by Spontaneous Parametric Down-Conversion of photons as they pass through a Type II crystal. Our main approach here will be phased-encoded BB84 running at a wavelength suitable for installed telco fibers (1550 nm).
QKD Switch	3	Team	A passive optical switch suitable for establishing, maintaining, and tearing down virtual circuits for single QKD photons, bounce by bounce, through a mesh of such switches from a source QKD Endpoint to an arbitrary destination QKD Endpoint.
QKD Circuit Management	3	Team	A suite of novel protocols and algorithms that will be implemented in the QKD Endpoints and QKD Switches to enable virtual circuit setup, maintenance, and teardown for frames of QKD photons.
Mirror Control	3	N/A	Device-control software to position and control the MEMS mirrors (or equivalents) within QKD Switches so that QKD photons reflect properly from an input fiber to the correct output fiber.
Mirror Array	3	Purchase	A passive optical switch, without amplifiers, suitable acting as the MEMS mirror array (or equivalent) within a QKD Switch. We expect that we will demonstrate with a relatively small array, e.g., one suitable for a 4 x 4 switch.
Ultra-Long Distance Fiber	4	Purchase	A novel form of optical fiber that offers much lower loss than today's conventional telco fibers, and which thus will enable QKD transmission over much longer distances than are possible with conventional fibers. A leading candidate at present is the hollow fiber from Dr. Yoel Fink's team at MIT, though other candidates may also present themselves as time goes on. Our plan here is to simply purchase such fiber and then to deploy and test it within the DARPA Quantum Network.
QKD Repeater	N/A	Unknown	The QKD Repeater is that semi-mythical beast, the quantum repeater. If developed, it would be very useful in the DARPA Quantum Network to help extend its geographic range. One teammate, Boston University, will perform research that may someday lead to a practical quantum repeater. Other teams around the world are engaged in similar pursuits. If any of these research projects are successful, we will attempt to duplicate the result and install it as a repeater in the DARPA Quantum Network.

9.2 External System Interfaces

This section defines the important external system interfaces for the DARPA Quantum Network. Note that details such as electrical power requirements, operating temperatures, and so forth, will not be discussed in this document as they are not of any great relevance for a research project. Figure 9-2 depicts the DARPA Quantum Network's important external system interfaces.

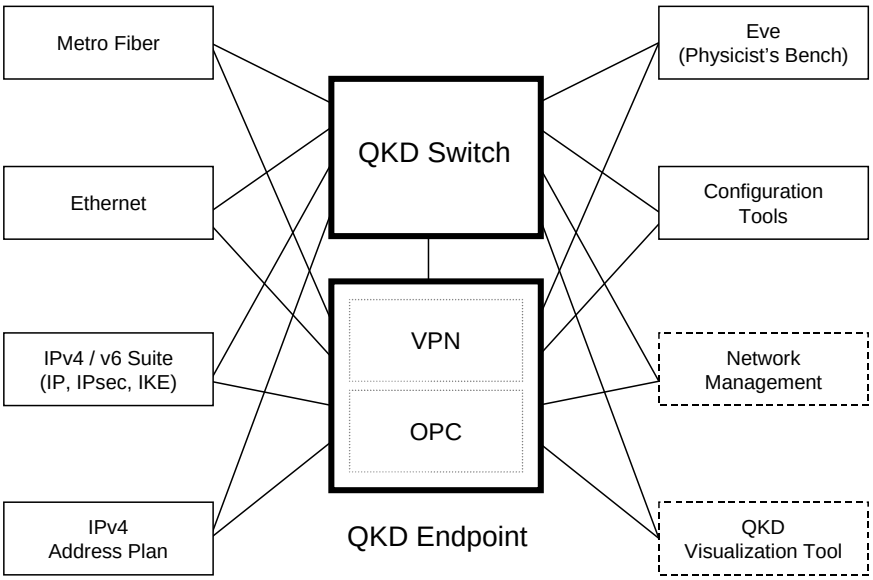


Figure 9-2. External System Interfaces for the DARPA Quantum Network.

Three of these external interfaces act as major design drivers for the DARPA Quantum Network. First, the network is designed to act as an Internet (IP) communications device and not, for instance, an ATM device or a telephone system device. Second, the network is designed to transmit QKD photons over today's installed base of telecommunications fiber, and in particular of fiber that is commonly present in metropolitan areas. This strongly influences the optical wavelengths employed in the QKD source and receiver suites. Third, we wish to grant our experimental Eve access to all interesting parts of the QKD system to facilitate eavesdropping experiments. Thus we can present certain interfaces externally in ways that we would not do if we were not trying to support eavesdropping experiments.

The following table briefly describes each external item in Figure 9-2.

External Entity	Description of Interface with Quantum Network
Metro Fiber	Conventional optical fiber as deployed for telecommunications within a metropolitan region. At present, QKD photons will be carried on a dedicated fiber; i.e., the QKD transmission fiber is not shared with any other sort of traffic. We require two conventional low-loss transmission windows (at 1300 nm and at 1550 nm) but do not require the broad windows designed for Dense Wavelength Division Multiplexing (DWDM) systems.
Ethernet	The QKD Endpoints and QKD Switches connect to external

	communications networks (the red and black Internets) via standard 100 Mbps Ethernet. Note that it would be easy to change this to any other Internet-conformant network interface if so desired.
IPv4 / IPv6 Suite (IP, IPsec, IKE)	The QKD Endpoints and QKD Switches must be fully compatible with the current Internet Protocol suite (IPv4), with the sole extension that they also implement a new dialect of IKE that has been augmented to take advantage of QKD technology. This version of IKE must be fully backwards compatible with existing standards. It is desirable, but not required, that the QKD Endpoints and QKD Switches also work with the next-generation Internet Protocol suite (IPv6).
IPv4 Address Plan	The QKD Endpoints and QKD Switches must support the following forms of IPv4 address plans: both red and black Internets use public IP addresses; red Internet uses private IP addresses but black Internet is public; both red and black Internet use non-public IP address plans.
Eve (Physicist's Bench)	Over time, we will implement many different versions of Eve, ranging from a simple spectrum analyzer to a complete set of QKD receivers, transmitters, and QKD protocol engines. Since a major portion of our research work is to investigate attacks on the DARPA Quantum Network, it is essential that Eve be capable of full access to the fiber between QKD transmitter and receiver, and cleartext versions of all messages among QKD Endpoints and QKD Switches in the DARPA Quantum Network. In short, it must be possible to experiment with an Eve that can readily break conventional cryptography (by turning such cryptography off in the QKD network devices), that can (in simulation) store qubits for extended periods of time, that can (in simulation) substitute lossless fibers for portions of the fiber between QKD Endpoints, and so forth.
Configuration Tools	TBD.
Network Management	TBD.
QKD Visualization Tool	TBD.

9.3 Major System Components and Interfaces

We now turn to descriptions of the major internal system components within the DARPA Quantum Network and of the interfaces between these components.

Figure 9-3 portrays the major internal system components as rectangular boxes. There are four such components in our network: IPsec (IKE), QKD Protocols, Optical Process Control, and Optical & Electronic Hardware. Each is described in summary below, and then in much greater detail in subsequent sections of this document. More components will be added in subsequent years, such as Internet-Level Routing (for eavesdropping-aware routing) and a cryptographic quality source of random numbers.

In Years 1 and 2, there is only one form of QKD link – weak-coherent with phase encodings – but in Year 3 we will add the BU entangled source as a second form of QKD link. Note that the overall system architecture will not be changed by the additional of the BU entangled source or indeed other new sources or detectors. Instead, these are simply new instances of the Optical & Electronic Hardware component and perhaps also of the Optical Process Control component as needed.

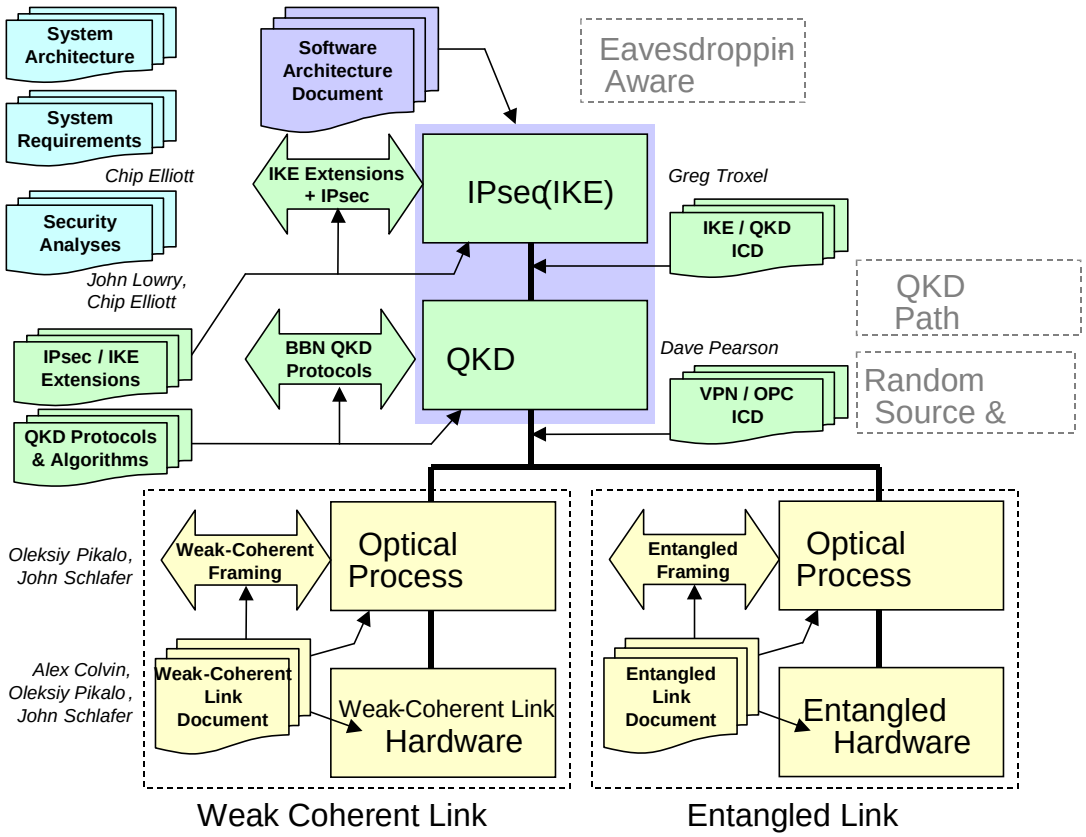


Figure 9-3. Major System Components and Interfaces.

Figure 9-3 also depicts the interfaces between major system components and indicates which documents describe each of these interfaces. There are two sorts of such interfaces: "horizontal" interfaces that define communication between similar peers in two different QKD Endpoints, and "vertical" interfaces that describe communication between different kinds of components within a single QKD Endpoint. Horizontal interfaces by documents to the left of the diagram; vertical interfaces are defined by Interface Control Documents (ICDs) shown at the right of the diagram.

Components. The following table briefly describes each of the major system components, as shown in Figure 9-3, and provides a reference to the section in this System Architecture Description that supplies a greater level of detail for that particular component.

System Component	Description
IPsec (IKE)	The IPsec (IKE) component consists of all portions of the QKD Endpoint that implement the IPsec protocol suite. In particular, this is a large collection of distinct software entities that run within both the kernel and user-space portions of the VPN computer in order to implement IPsec. This software comes as-is with the operating system in the VPN computer. The Internet Key Exchange (IKE) software entity is an important part of this collection since it is responsible for establishing, maintaining, and tearing down IPsec security associations. Our team will modify this software as necessary in order to augment IKE so that this extended version of IPsec can take advantage of quantum cryptography. See Section 16 for details.

QKD Protocols	The QKD Protocols component consists of the software entity that implements all the QKD protocols and algorithms in the DARPA Quantum Network. This software will be custom-developed for the DARPA Quantum Network. See Section 13 for details.
Optical Process Control	The Optical Process Control (OPC) component is a software entity that performs two distinct tasks vis a vis the Optical and Electronic Hardware. The first task is to monitor and control the housekeeping aspects of this hardware, e.g., adjust the laser temperatures as needed, continuously monitor and control the path lengths for the Mach-Zehnder interferometers, etc. The second task is to provide the "data path" for the QKD photons, i.e., establish the (basis,value) pairs for transmission, trigger the source lasers, sample the detectors at the receive suite, etc. This second task also involves the establishment of framing information for the QKD link. Note that the OPC software entity is housed on its own dedicated computer, called the OPC computer. See Section 10 for details.
Optical and Electronic Hardware	The Optical and Electronic Hardware is a hardware entity that implements a transmitter or receiver suite for QKD photons. On the transmit side, it consists of devices such as lasers, attenuators, interferometers, phase modulators, and the electronic equipment that drives them. On the receive side, it consists of devices such as polarization controllers, phase modulators, and cooled detectors, along with the attendant electronics. See Section 10 for details.
Eavesdropping-Aware Routing	This will be designed and implemented in Contract Year 3. See Section 16 for details.
QKD Switch Path Control	This will be designed and implemented in Contract Year 3. See Section. See Section 10 for details.
Random Number Source and Tests	The Random Number Source and Tests provide a sequence of unpredictable bit values that are used in many other subcomponents. See Section 11 for details.

Vertical Interfaces. The following table briefly describes each of the major vertical system interfaces, as shown in Figure 9-3, with references to more detailed description within this document.

System Interface	Description
IKE / QKD ICD	Software interface between the Internet software entity that performs the Internet Key Exchange (IKE) protocol, and the software entity that performs the QKD protocols and algorithms. This interface allows the IKE entity to reserve and obtain frames of shared secret bits from the QKD entity, as well as perform a variety of housekeeping operations. See Sections 9.5.2 and 13.14 for more details.
VPN / OPC ICD	Software and hardware interface between the software entity that performs the QKD protocols and algorithms, running in a VPN computer, and the Optical Process Control (OPC) software entity running in its own computer. In essence, the OPC entity sends a continuous series of frames representing QKD photons as transmitted or detected, along with a frame number for each frame. This interfaces also defines a variety of housekeeping operations by which the QKD entity may control the OPC entity. See Sections 9.5.1 and 10.9 for details.

Horizontal Interfaces. The following table briefly describes each of the major horizontal system interfaces, as shown in Figure 9-3, with references to more detailed description within this document.

Protocol	Description
IKE and IPsec Extensions	Extensions to the standard Internet Key Exchange (IKE) protocol. These extensions will be backward compatible with the existing IKE standard, and will be documented in the "IKE Extensions Document" in a manner suitable for submission to the Internet Engineering Task Force (IETF), which is the standards body for the Internet protocols, as either a research contribution or possibly a standards-track document.
BBN QKD Protocols	A complete suite of QKD protocols and algorithms, to include all the mechanisms necessary to implement BB84, B92, sifting, Cascade (BS92) error correction, simpler error correction schemes as devised, and privacy amplification. All such protocols and algorithms will be documented within a single compendium (the "Protocol and Algorithm Definition Document").
Weak-Coherent Framing	Definitions of the QKD single-photon link for both Layer 1 (physical) and Layer 2 (framing). This interface defines the wavelengths of both bright annunciator pulses and weak, single-photon pulses, along with their timing requirements. It also defines frame formats for this link along with the state machines necessary for obtaining frame synchronization and detecting loss of such synchronization. This interface will be documented within the overarching "Photonics Subsystem Document."

9.4 Major Components and Data Flow in the "Trusted" QKD Network

This section provides a technical introduction to the major components (sub-systems) and data flows within the "trusted" version of the Quantum Network. It does not describe how photonic switches will work in the "untrusted" Quantum Network; see Section 7.7for a discussion of those switches.

Figure 9-4 displays a protocol-stack schematic of how a "trusted" link fits into the Quantum Network. As shown, each QKD Endpoint is currently implemented as a pair of computers – a Virtual Private Network (VPN) computer that implements the QKD Protocols and the Internet Protocol suite, and an Optical Process Control (OPC) computer that manages the opto-electronic equipment that implements a Weak Coherent key distribution link. These two computers, VPN and OPC, are joined by a private Ethernet that lets data and control messages flow between them.

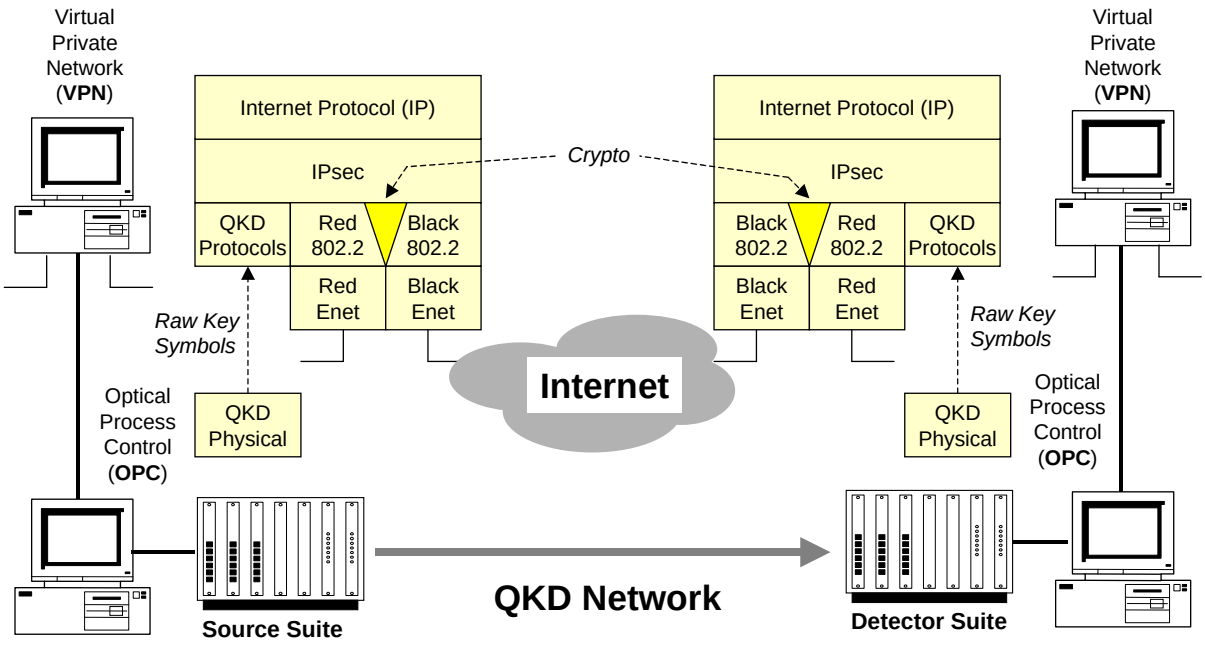


Figure 9-4. High-Level Overview of a "Trusted" Link in the Full Network Context.

It is important to note that the QKD link is *not* a communications link. It is used solely for key agreement, and not for data traffic. Hence it does not quite fit into a conventional network protocol stack diagram. Close inspection of Figure 9-4 reveals that data flows *upward* only from the OPC to its VPN twin, at both the Source and Detector side of the QKD link. This data is not message traffic but is rather information about raw qubits as transmitted and detected; after processing it will turn into keying material at both QKD Endpoints.

Figure 9-5 shows the major flows of data within a single trusted link, i.e., between two QKD Endpoints. This diagram reveals a great many internal details about our implementation of the QKD Endpoints, which are described in detail in subsequent sections. For now, we shall concentrate on the high-level descriptions of the data flows within a trusted link and the interlocking sets of protocols that enable these data flows.

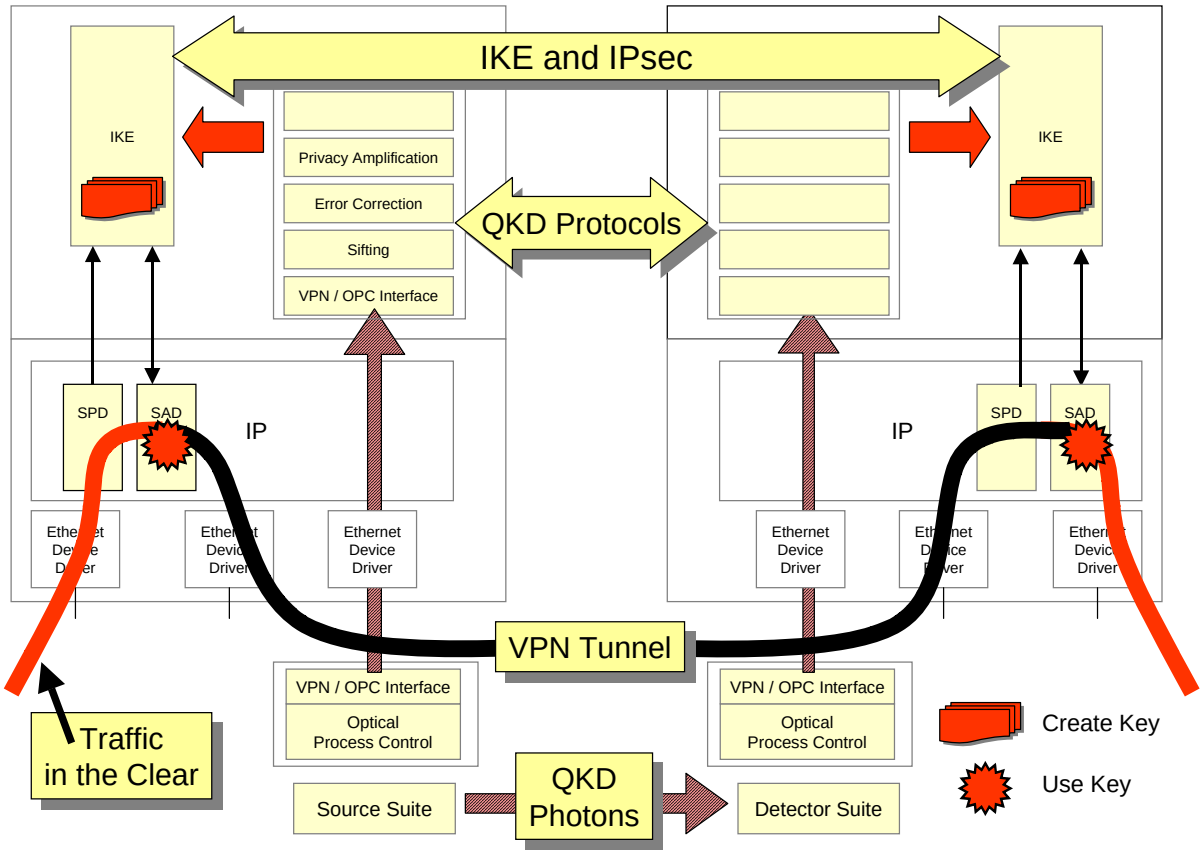


Figure 9-5. Protocols and Data Flow within a "Trusted" Link in the Full Network Context.

A brief reminder of the basic principles of operation: a) two QKD endpoints establish communications via a dedicated fiber or wavelength for the quantum path, and via the Internet for messaging; b) the transmit side prepares and transmits raw keys, from which both sides come to agreement on a shared, secret key; c) this secret key is then employed in the VPN cryptographic gateway for protecting message traffic that will transit the Internet within secured IPsec tunnels.

Figure 9-5 documents a single link in the DARPA Quantum Network during routine operation. At such time, the VPN and OPC computers operate in only loose synchrony. In essence, the OPC source and detectors suites provide a continuous stream of raw key symbols to their corresponding VPN computers. These symbols include both the basis used for a given bit (as transmitted or received) and the bit's value, along with QKD framing information. As the VPN computers receive these frames of raw key symbols, they perform a suite of well-known QKD protocols (sifting, error correction, privacy amplification, etc.) to derive the actual "good" key bits. These good bits continuously accumulate in a reservoir of shared keying material within each QKD endpoint. This keying material is then available for use via the IPsec protocol suite where it can be used to encrypt one or more secure VPN tunnels through the Internet. In particular, the keying material is fed to an encryption algorithm (crypto) for use in encrypting or decrypting IP datagrams as they pass through the VPN computer. Then as traffic flows enter the VPN computer in the clear from the private enclave (conventionally known as the "red" side of the gateway), they pass through this crypto and become encrypted. These encrypted datagrams are then carried through the "black" public Internet, or any other suitable communications network, until they are received at the

distant VPN gateway, where they are passed through another crypto and thus decrypted, and then sent once again in the clear onto the "red" private enclave at the distant location.

Restating the preceding paragraph in greater technical detail, the major data flows in Figure 9-5 are:

- The data traffic enters a QKD Endpoint "in the clear" (unencrypted) via its Red Ethernet, e.g., from the left edge of Figure 9-5. It becomes encrypted and authenticated as it passes through the QKD Endpoint, and transits the Internet in this "black" (encrypted) form through a VPN Tunnel until it is received at the peer QKD Endpoint and decrypted. The specific locus of the encryption and decryption algorithms is the SAD entity within the kernel of each QKD Endpoint, which is the Security Association Database component within the IPsec suite. This SAD contains and employs a per-session key that is used to protect traffic within a given secured flow.

- Asynchronously, the Weak Coherent link is creating and transmitting raw frames of QKD qubits from the Source side to the Detector side. Each such raw frame, called a Qframe, is sent upwards from the OPC computer at the Source or Detector to its attached VPN computer. See Section 10 for detailed information on the weak coherent link.

- Within each VPN computer, this raw Qframe is processed and turned into a shared set of secret, random bits. This processing requires communication between peer VPN computers and in particular between the QKD Protocol Daemons in each. These communications are termed the "QKD Protocols" and indeed an entire suite of protocols is required instead of a single protocol. These protocols allow the sifting of raw Qframes to remove bits detected in the wrong basis, error correction of the sifted bits, privacy amplification, and so forth. See Section 13 for detailed information on the QKD protocols and algorithms.

- The shared secret bits are transported from the QKD Daemon in each VPN computer to its IKE Daemon for use in IPsec key agreement protocols. In particular, the two IKE peers use an extended variant of the standard IKE protocol to agree on the shared secret bits that they will employ as the basis for encryption and authentication of each VPN Tunnel. Note that the shared secret bits are never explicitly conveyed between the IKE peers; instead they convey identifiers for blocks of shared secrete keys (Qblocks) that are maintained in the parallel QKD Daemons.

- When the peer IKE Daemons agree upon a set of shared secret bits to employ, they then fetch the corresponding Qblocks from their respective QKD Daemons and process these bits to obtain per-session keying material for a given VPN Tunnel. As usual in IKE implementations, the IKE Daemons then pass these keys downward into the operating system kernel for use in the SAD as it processes data traffic for this VPN Tunnel. See Section 16 for detailed information on the IPsec and IKE mechanisms.

The following table provides a brief description of the various entities within Figure 9-5.

Element in Figure 9-5	Description
IKE	The software entity that implements our augmented version of the Internet Key Exchange (IKE) protocols and algorithms. (BBN modification of the IKE Daemon supplied by KAME.)

Authentication	TBD.
Privacy Amplification	Software that implements QKD privacy amplification protocols and algorithms. (BBN-supplied software resident in the QKD Daemon.)
Error Correction	Software that implements QKD error correction protocols and algorithms. (BBN-supplied software resident in the QKD Daemon.)
Sifting	Software that implements QKD sifting protocols and algorithms. (BBN-supplied software resident in the QKD Daemon.)
VPN / OPC Interface	Software that implements the interface defined in the "VPN / OPC ICD" for communication between the QKD protocol stack and the Optical Process Control entity. (BBN-supplied software resident in both the QKD Daemon and in the OPC entity.)
SPD	Security Policy Database. A database together with algorithms that classify IP datagrams to determine which datagrams belong in which security associations. This is done by pattern-matching of various fields in the IP datagrams with rule sets in the database. If a datagram is found that requires a security association but does not yet have one in place, a signal is sent from the SPD to the IKE Daemon requesting that such an association be established.
SAD	Security Association Database. A database together with algorithms that perform IPsec actions on IP datagrams as needed for a given security association, e.g., encryption or decryption, authentication, encapsulation, and the like. This is where the "crypto" is implemented in IPsec.
IP	The software entity that perform Internet Protocol (IP) datagram forwarding, including all IPsec operations. This is resident in the operating system's kernel.
Ethernet Device Driver	The software entity that controls the Ethernet network interface card in a computer. This is resident in the operating system's kernel.
Optical Process Control	The software entity that controls the opto-electronic QKD transmitter or receiver suite and that imposes framing on the QKD link. This is the software entity responsible for continuously transmitting or receiving frames of QKD symbols and reporting the results to the QKD Protocol software entity. (BBN-supplied software resident in the OPC computer.)
Source Suite	The opto-electronic hardware suite that implements the transmitter (source) side of a QKD link. (BBN-supplied hardware equipment string.)
Detector Suite	The opto-electronic hardware suite that implements the receiver (detector) side of a QKD link. (BBN-supplied hardware equipment string.)
Create Key	Location where session keys are created. They are created in the IKE Daemon based on distilled key bits received from the QKD protocol stack.
Use Key	Location where session keys are used. They are used in the "crypto" algorithm resident in the IPsec part of the IP forwarding engine in the VPN computer's operating system kernel. We have represented this as taking place in the SAD.
Traffic in the Clear	"Red" traffic, i.e., that which is freely exchanged in clear text form within a private enclave. This traffic will pass through a crypto and become encrypted before it is injected into the "black" (public, untrusted) Internet for delivery to the destination enclave. At the destination VPN gateway, it will be decrypted, authenticated, and integrity-checked, and again injected into

	the destination-side private enclave as red traffic.
VPN Tunnel	A security association between two VPN gateways so that traffic flowing from one of these gateways to the other is handled according to the rules for this particular security association. In general, such VPN tunnels are encrypted, authenticated, protected against replay attacks, and encapsulated to hide the IP addresses of the source and destination machines.
QKD Photons	Single photons sent along a specialized fiber so as to implement the physical layer of the QKD protocols.

Figure 9-6 depicts the same basic information within a broader network context. Here we depict three QKD Endpoints in a full trusted Quantum Network, rather than just a single link. We have shown a random assignment of sources and detectors to each of the Alice, Bob, and Charlie nodes. Note that it is not important whether a given node contains a Source or a Detector for any given QKD link. As shown, each QKD link is managed by a its own OPC computer, which reduces the complexity required for the LabView program in each OPC. However, each VPN computer must manage an indeterminate number of OPCs (i.e. multiple QKD links), one for each QKD peer.

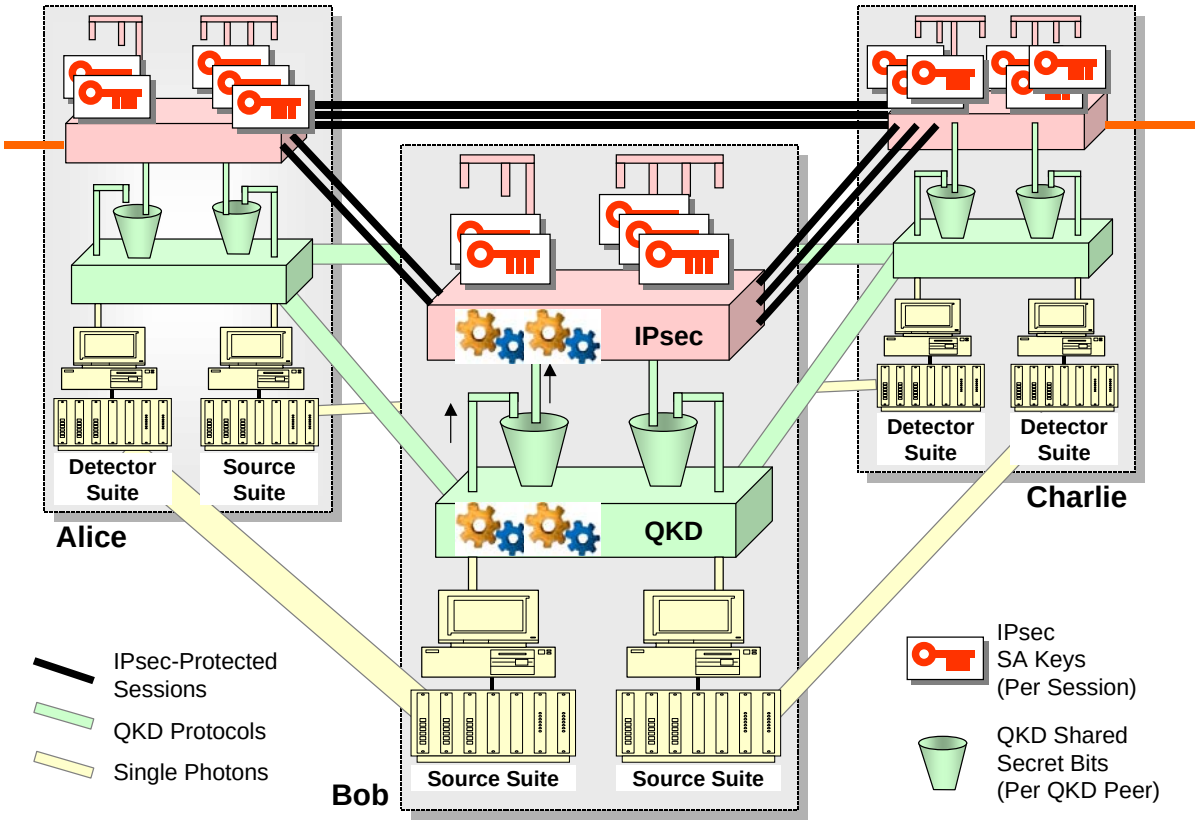


Figure 9-6. Major Components in the "Trusted" Quantum Network.

Element in Figure 9-6	Description
Alice, Bob, Charlie	QKD Endpoints, each of which has a Red Ethernet and a Black Ethernet for message traffic, and one or more QKD links for quantum key distribution with its peers.
IPsec	A version of the IPsec (and IKE) protocol suite, extended to manipulate key material obtained by QKD techniques.
QKD	BBN QKD Protocols and algorithms, e.g., sifting, error correction, privacy amplification, etc.
Source Suite	An Optical Process Control (OPC) computer, together with all the opto-electronic equipment required to act as the Source side of a Weak Coherent QKD link. For Year 1, this includes one or more lasers, attenuators, Mach-Zehnder interferometer, phase modulator, etc.
Detector Suite	An Optical Process Control (OPC) computer, together with all the opto-electronic equipment required to act as the Detector side of a Weak Coherent QKD link. For Year 1, this includes a Mach-Zehnder interferometer, phase modulator, two cooled APD detectors for the BB84 protocol, etc.
IPsec-Protected Sessions	Each IPsec-protected session is a separate set of one or more traffic flows that have been grouped together under a single Security Association. In most cases, these will be used as a full Virtual Private Network (VPN) tunnel with encryption, authentication, etc.
QKD Protocols	BBN-supplied protocols, running over IP, that implement the QKD protocols needed for sifting, error correction, privacy amplification, etc.
Single Photons	In Year 1, the single photons are generated by a highly attenuated telecommunications laser at 1550 nm, pulsed at 5 MHz. This implements a classic Weak Coherent QKD link.
IPsec SA Keys	Cryptographic keys for a Phase 2 Security Association (SA) between two VPN gateways that carries message traffic for a particular Virtual Private Network traffic flow.
QKD Shared Secret Bits	A set of one or more Qblocks, where a Qblock is a fixed-size block of shared secret bits based on the contents of Raw Qframes as refined by the QKD protocols. These bits are always sifted and error corrected; they may also be privacy amplified and/or authenticated, depending on system configuration. (In operational use, we expect that privacy amplification and authentication would be enabled.)

Figure 9-6 highlights the relationships between QKD links, OPC peers, QKD peers, and VPN Tunnels. These relationships can be summarized as follows:

- Each QKD link functions autonomously and asynchronously (from each other, and from other portions of the overall system) to deliver sequences of raw Qframes to its attached VPN computer. Note in particular that there is no shared clock across a set of QKD links; instead each has its own clock.

- The raw Qframes for a given QKD link are processed by the associated QKD Daemon for that link. Note that a single QKD Daemon may manage multiple QKD links, however. Thus it must keep

separate databases for the Qframes from each link, as well as manage multiple instances of the QKD protocols.

- Because of randomized choice in photon basis selection, bursts of noise, and variable delay in the operation of the QKD protocols, the QKD algorithms will produce distilled secret bits at a variable rate. These distilled secret bits are stored locally within the QKD Daemon's memory in a structure called a Qblock. (This storage is depicted as a bucket in Figure 9-6 in order to emphasize that purified bits "slosh in" a variable rate, and are drawn out as needed by IKE in an asynchronous manner.) The QKD Daemon keeps a pool of Qblocks for each of its QKD peers, but the Qblocks for one peer are never mingled with those for another.

- Finally, the IKE Daemon in a given QKD Endpoint will fetch Qblocks as needed from its associated QKD Daemon, and use the bits in these Qblocks to create one or more session keys. These session keys are created on a per-peer basis. That is, IKE will only create an IPsec security association with a direct QKD peer, because it can only use QKD bits shared with that peer.

9.5 Major Interfaces within a QKD Endpoint

This section introduces the major interfaces within a QKD Endpoint, and shows – at a high level – the types of data that flow across these interfaces. Each of these interfaces is described in somewhat more detail in subsequent portions of this document, and each is defined by its own Interface Control Document (ICD) which provides a highly-detailed definition of the relevant interface. The two major interfaces are:

- The VPN / OPC interface, which lies between the QKD Protocols in the VPN computer and the LabView program running in the OPC computer. See Section 9.5.1 for a further level of detail on this interface, or the "VPN / OPC ICD" for the authoritative definition of this interface.

- The IKE / QKD interface, which lies between the IKE Daemon and the QKD Daemon, both running in a single VPN computer. See Section 9.5.2 for a further level of detail on this interface, or the "IKE / QKD ICD" for the authoritative definition of this interface.

Figure 9-7 depicts these two interfaces within the context of the overall QKD Endpoint internal architecture. As shown, the VPN / OPC interface is implemented by a set of C-language calling sequences that abstract away the underlying Ethernet link between the two computers.

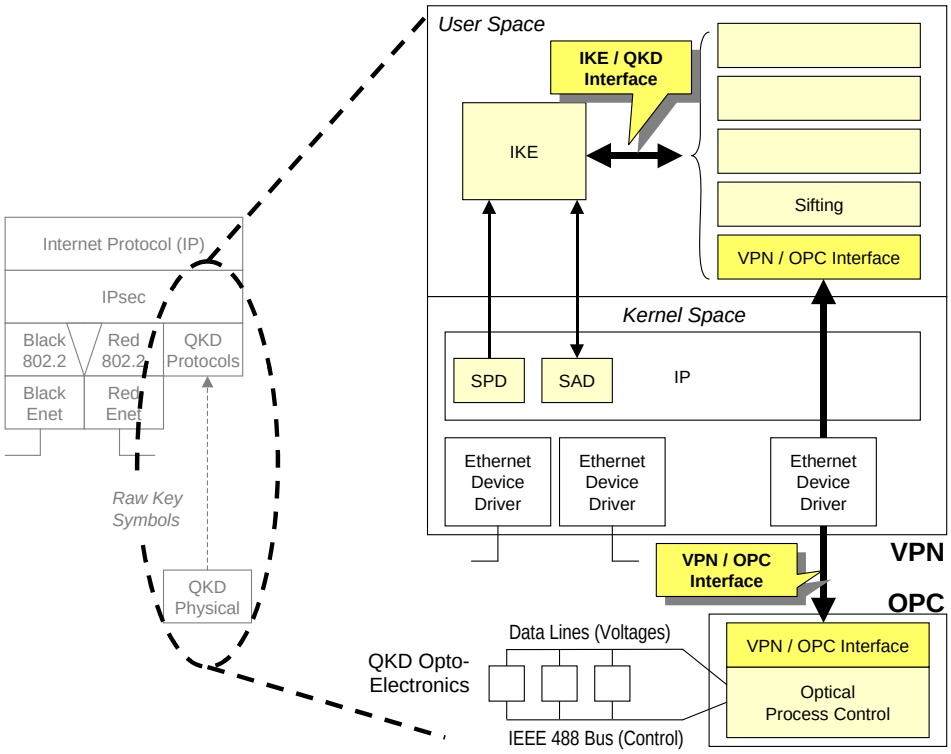


Figure 9-7. Major Internal Interfaces within a QKD Endpoint.

9.5.1 The VPN / OPC Interface

The VPN / OPC interface is arguably the most "public" design point in our overall system architecture, because it defines the attachment point for equipment strings that will be supplied by a number of different QuIST teams, i.e., their opto-electronic equipment and associated control computers.

Figure 9-8 shows this interface in cartoon form. As can be seen, all the "protocols" are implemented in a Virtual Private Network (VPN) device, and all the "optical control" is implemented in a separate Optical Process Control (OPC) computer. The VPN / OPC interface thus defines what these two computers say to each other and how they say it.

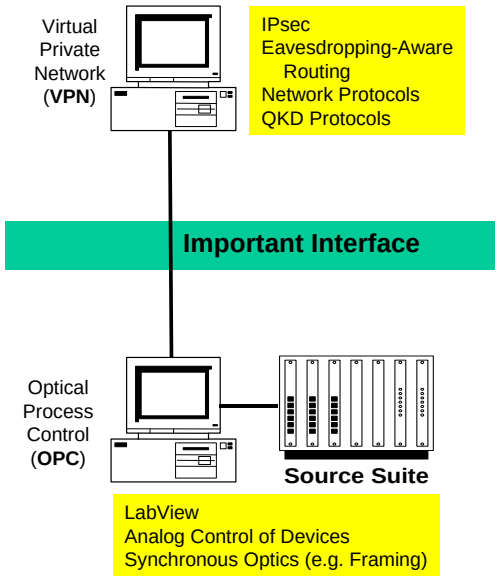


Figure 9-8. Interface Between Networking and Photonics Subsystems in a QKD Endpoint.

We have chosen to split a QKD Endpoint into two distinct computers for the following reason. After reading a number of published articles on quantum cryptography, and talking with some of the implementers, it appears to us that most (or perhaps all) of the research community uses LabView running in a dedicated computer to control their QKD equipment. Furthermore, most researchers seem to prefer running LabView under the Windows operating system. For a variety of reasons, we believe it is best to run the IPsec protocol suite under NetBSD, a Unix-like operating system, so already we have a slight clash of cultures.

Since we strongly desire to make it easy for other research teams to bring their equipment strings into the Quantum Network, it seems wisest to explicitly allow other research teams to keep using LabView under Windows in the Quantum Network. Hence we have designed our overall architecture to make this not only feasible, but easy. All the networking protocols go in one (Unix) computer; all the optical control goes in another (Windows) computer.

These two computers – VPN and OPC – are linked by a private, 100 Mbps Ethernet. A specialized set of BBN-supplied protocols, with C language calling interfaces, allows easy communication between the two computers. One side of this interface is implemented in the QKD Daemon in the VPN computer; the other is integrated with LabView in the OPC. While this interface is not precisely “plug and play” for new equipment entering the Quantum Network, we believe that the cost of integrating this C code into another team’s LabView program will be minimal. Note that system security depends on this link being truly private.

9.5.2 The IKE / QKD Interface

The other major interface within a QKD Endpoint lies between the QKD Protocols and IPsec. More precisely, in our implementation, it is an interface between our QKD Daemon running in application space, and the IKE Daemon also running in application space on the same VPN computer.

Figure 9-9 depicts the flow of data across these major interfaces. In particular, it introduces the precise terminology we use for each kind of "key" material in the system: raw Qframes rising up from the optical layer, Qblocks of secret shared bits that have been produced by the QKD protocols, and the Session Key used to protect an individual IPsec tunnel.

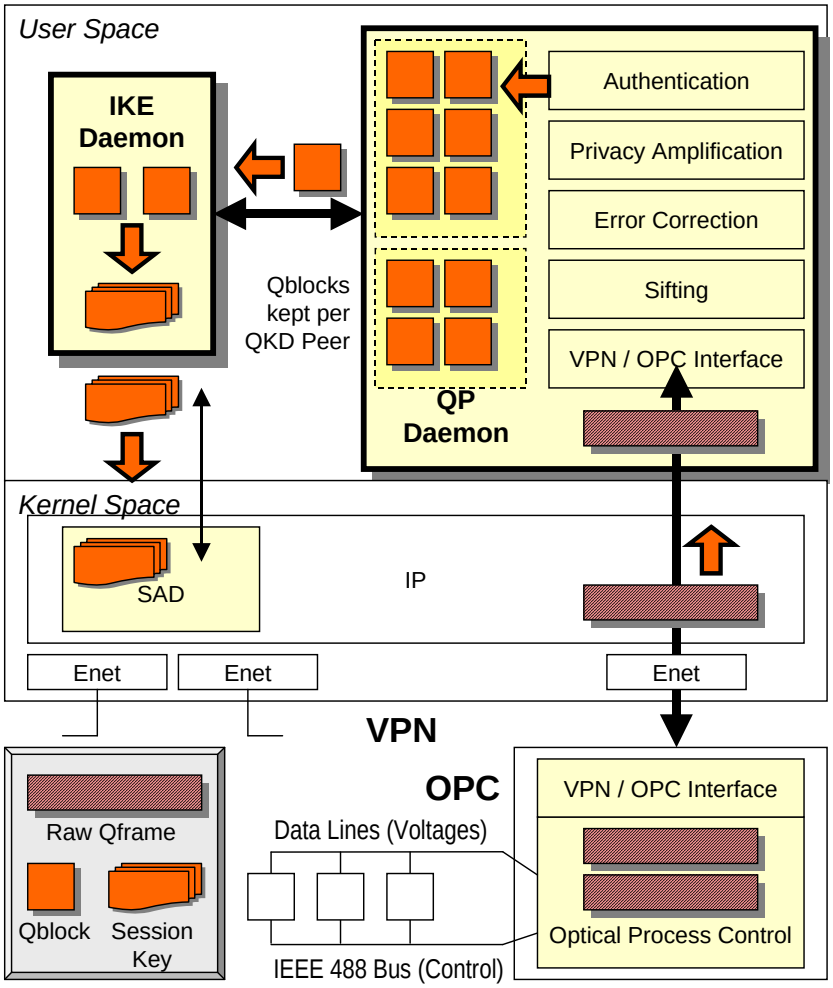


Figure 9-9. Relationship of Qframes, Qblocks, and IPsec Keys in a QKD Endpoint.

Entity	Description
IKE Daemon	The software entity that implements our augmented version of the Internet Key Exchange (IKE) protocols and algorithms. (BBN modification of the IKE Daemon supplied by KAME.)
QP Daemon	The software entity that implements BBN's version of the Quantum Cryptographic Protocol suite, including all QKD protocols and algorithms running in the DARPA Quantum Network.
Raw Qframe	A Raw Qframe is a fixed-size block of symbols transmitted from an OPC to

	its VPN. At the transmit side, it contains an indication of the bases and values that were prepared for each outbound single-photon qubit. At the receive side, it contains the basis and detector-hit values for each inbound qubit. See Section 10.6 for a description of how Raw Qframes reflect the underlying physical processes on the QKD link. The “VPN / OPC” Interface Control Document defines the exact format of Raw Qframes.
Qblock	A Qblock is a fixed-size block of shared secret bits, produced by the QP Daemon for use by the IKE Daemon and based on the contents of Raw Qframes as refined by the QKD protocols. These bits are always sifted and error corrected; they may also be privacy amplified and/or authenticated, depending on system configuration. (In operational use, we expect that privacy amplification and authentication would be enabled.)
Session Key	A session key is a cryptographic key that is used for a single “session” and then discarded. In the DARPA Quantum Network, a session is defined to be a certain traffic flow (or set of flows) for a given duration of elapsed time or a given number of traffic bytes. The set of flows, and the choice between duration or traffic bytes, is governed by system configuration.

9.6 The Relationships between Photonics, QKD Protocols, and IKE

Figure 9-10 diagrams how the various levels of protocols interact within the overall system framework of a single QKD link in the Quantum Network.

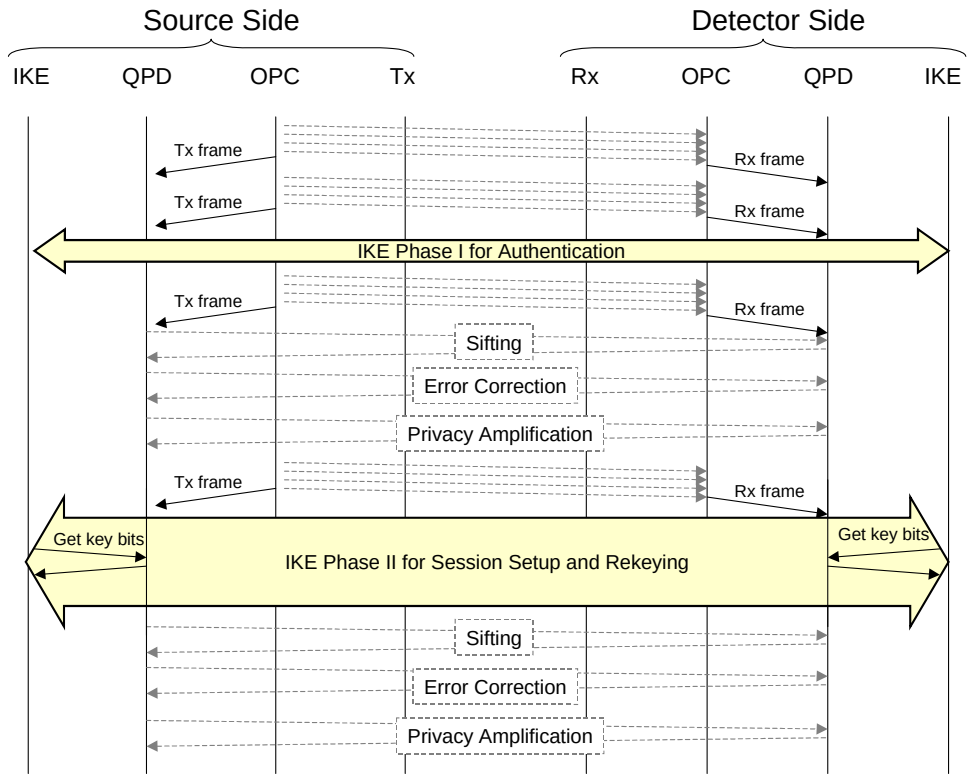


Figure 9-10. Relationships between Photonics, QKD Protocols, and IKE.

As can be seen, there are four distinct layers of protocols at the source side (Alice) with corresponding layers at the detector side (Bob). These layers are the IKE Daemon which performs Internet-level security association management, the Quantum Protocol Daemon (QPD) which implements the QKD protocols, the Optical Process Control (OPC) equipment which sends both framing information and qubits from Alice to Bob, and the actual transmit (Tx) and receive (Rx) suites of opto-electronic equipment.

At the lowest two levels, all information flows in just a single direction across the link, namely from Alice to Bob. Thus both framing information and qubits are created at Alice and sent across the QKD fiber to Bob. At this layer, Bob does not send any information back to Alice. Furthermore, information flows only *upwards* from the OPC to QPD at both ends of the link. This information is Raw Qframes and will form the basis of the shared secret bits between Alice and Bob.

At the higher protocol layers, namely the QKD protocols and IKE, communication flows in both directions between Alice and Bob. This does not mean that it is symmetric; indeed, the BBN QKD protocols are distinctly asymmetric. However in both layers, Bob does indeed transmit information back to Alice.

Note further that Alice's OPC sends framed qubits to Bob's OPC at an almost-constant rate, no matter what else is happening in the system. As a result, each OPC accumulates Raw Qframes at a rate derived directly from this underlying transmission of qubit frames, and thus each OPC delivers a Raw Qframe to its QPD peer at a roughly constant rate. The private link between the OPC and QPD is inherently somewhat asynchronous, however, and indeed requires a reliable transport protocol to ensure that frames aren't accidentally dropped. Thus delivery of Raw Qframes to the QPD is not synchronous but in practice the frames should appear at a fairly even pace.

Any given set of Raw Qframes, however, will require a variable amount of processing before they are completely distilled. This processing may in general take a different number of roundtrips of the QKD protocols, and will produce a variable number of distilled secret bits after sifting and error correction, etc. Hence the flow rate of secret bits from the QPD to OPC is likely to be highly variable, despite the fact the at the lowest layers of the stack the single photons are being transmitted at a nearly constant rate.

Finally, note that at present we do not expect the IKE Daemon to employ any QKD shared bits in its Phase 1 interactions with its peer. However, it will request shared secret bits from its local QPD whenever it needs to set up a new security association (encrypted tunnel) for data traffic, or needs to refresh the key material for an existing security association. As shown, these local requests for secret QKD bits happen in the middle of such "Phase 2" IKE interactions.

10 The Mark 2 Weak-Coherent Link

This section describes the opto-electronic subsystem in the DARPA Quantum Network, which implements our Mark 2 Weak-Coherent QKD link, along with its Optical Process Control computer and software. It introduces the basic equipment string for the weak-coherent link, describes how optical framing works, and summarizes the basic functions of the Optical Process Computer. This section also describes the interface between the Optical Process Control (OPC) subsystem and the QKD protocol suite running in the Virtual Private Network (VPN) computer.

10.1 Overview of the Mark 2 Weak-Coherent QKD Link

Our first QKD link for the DARPA Quantum Network is a weak-coherent link employing the BB84 protocol and encoding qubit values in the phases of individual photons. This scheme was first proposed by Bennett in 1992 and has been implemented several times by various research teams. Our implementation is most strongly influenced by the Los Alamos system, though we have engineered our own version from the components up in order to take advantage of current technology.

Figure 10-1 highlights the major features of our weak-coherent link. As shown, the transmitter at Alice sends single photons by means of a very highly attenuated laser pulse at 1550.12 nm. Each of these photons passes through a Mach-Zehnder interferometer at Alice which is randomly modulated to one of four phases, thus encoding both a basis and a value in that photon's self interference. The receiver at Bob contains another Mach-Zehnder interferometer, randomly modulated to one of two phases in order to select a basis. The received single photons pass through Bob's interferometer to strike one of the two cooled detectors and hence to present a received value. Alice also transmits bright pulses at 1550.92 nm, multiplexed over the same fiber, to send timing and framing information to Bob.

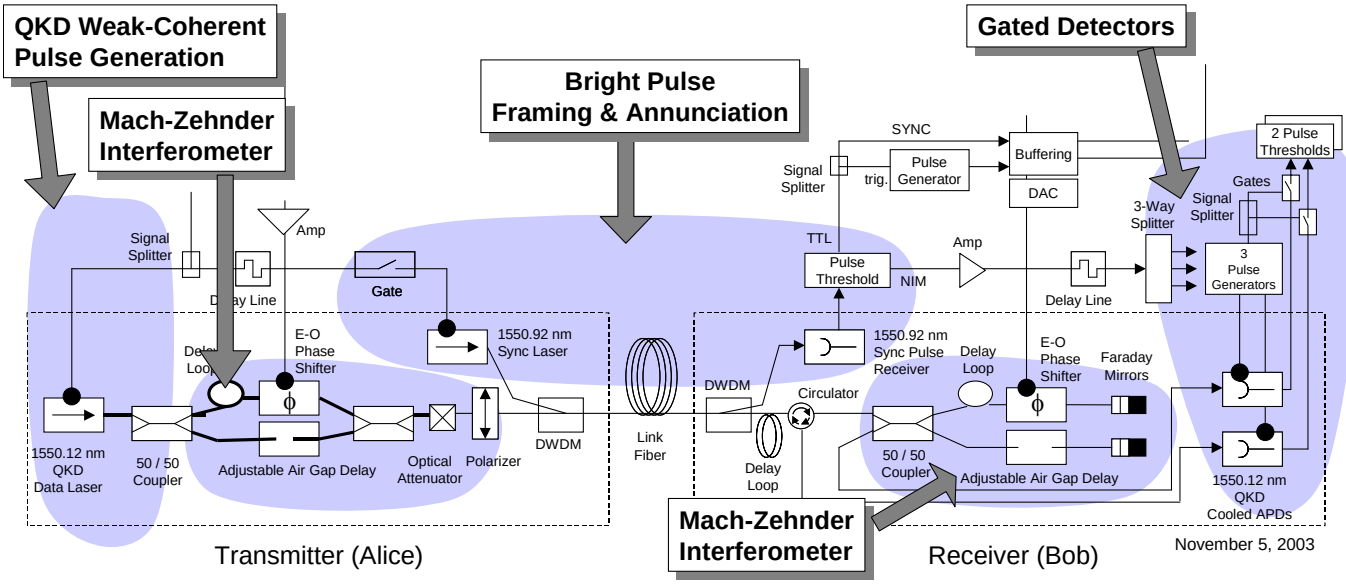


Figure 10-1. Functional Decomposition of the Mark 2 Weak-Coherent QKD Link.

Figure 10-2 illustrates the basic mechanism underlying our phase-encoding scheme for conveying bits via attenuated pulses[1]. As shown, Alice contains an unbalanced Mach-Zehnder interferometer, i.e., an interferometer in which the two legs have different delays. Bob contains a similar interferometer; in fact, certain dimensions of the two interferometers must be kept identical to within a fraction of the QKD photon's wavelength, i.e., a fraction of 1550 nm. We have labeled the various paths that a photon can follow through these interferometers for ease of discussion in the following paragraphs.

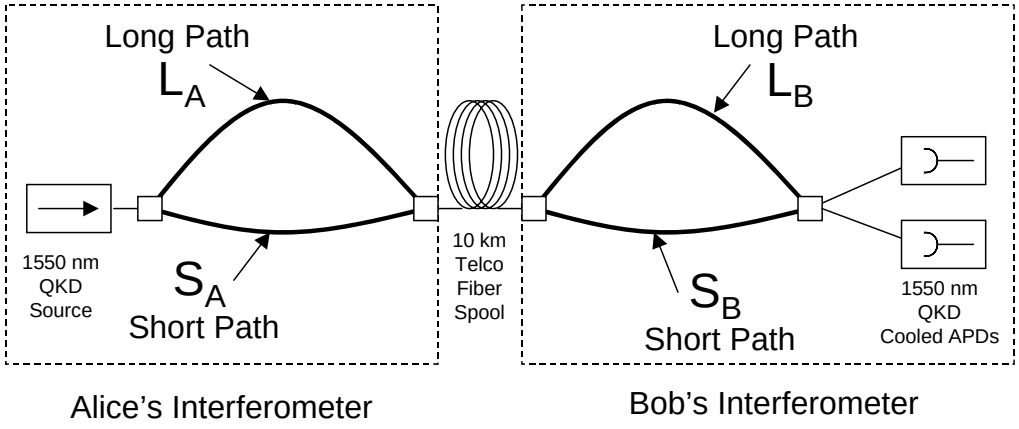


Figure 10-2. Path Components in Unbalanced Mach-Zehnder Interferometers.

Figure 10-3 shows how a single photon behaves as its pulse proceeds from the 1550 nm QKD source at Alice towards the pair of detectors at Bob. Here one should visualize the photon as a wave rather than as a particle. Thus it follows *both* paths of each interferometer rather than having to choose a single path. Not surprisingly, the part of the photon pulse that follows the shorter leg of an interferometer will emerge somewhat sooner than that part of the pulse that takes the longer leg.

Reading from the left of Figure 10-3, we see a single photon pulse emitted from the QKD source. It then follows both legs in Alice's interferometer and the part that follows the longer path (labeled LA) begins to lag behind that which takes the shorter path (SA). These two halves are combined at at the 50 / 50 coupler before they leave Alice and travel as two distinct pulses through the telco fiber loop.

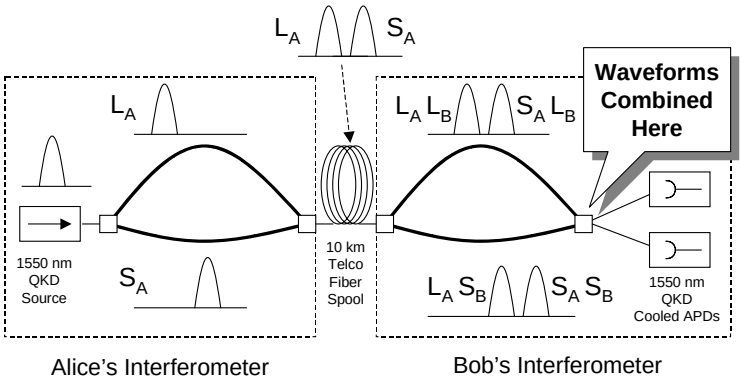


Figure 10-3. Effects of an Unbalanced Mach-Zehnder Interferometer on a Single Photon.

[1] This scheme was first proposed in C. H. Bennett, 1992, "Quantum cryptography using any two nonorthogonal states," Phys. Rev. Lett. **68**, 3121-3124.

Once this double-pulse photon reaches the interferometer at Bob, it once again takes both paths through Bob. Thus the part of the double pulse that takes the top path (long path) will be delayed relative to that which follows the shorter, bottom path.

Figure 10-4 shows how Bob's 50 / 50 coupler (just before the detectors) combines the resulting double pulses. If the interferometers are set correctly, the leading pulse in the upper train will align more or less precisely with the trailing pulse in the bottom train, and the two amplitudes will be summed. The right part of the diagram shows the resulting combined waveform at the 50 / 50 coupler just in front of Bob's pair of QKD detectors.

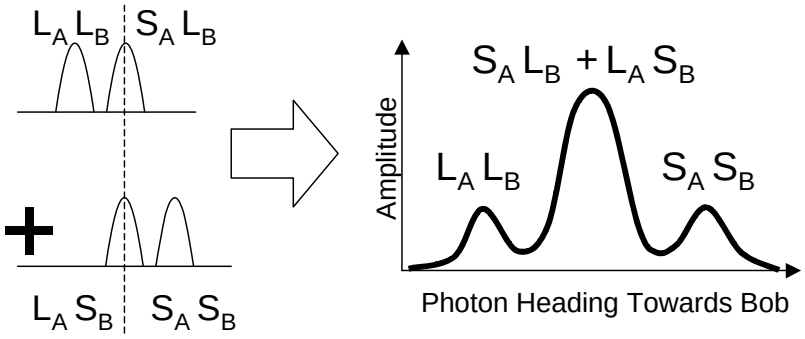


Figure 10-4. Recombined Photon at 50 / 50 Coupler just before Bob's QKD Detectors.

As a detail, as shown in Figure 10-5, Bob's detectors must be "gated" so that they are activated just around the time that the central peak in this photon arrives at the detectors. This means that a bias voltage must be applied to the detectors, just in time, and then turned off. Gating is not part of the quantum cryptography scheme as such; it is required only because we do not have reliable single-photon detectors and are somewhat misusing conventional detectors.

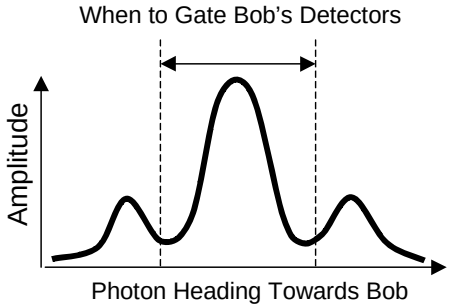


Figure 10-5. Gating Bob's Detectors to catch a QKD Photon.

We are now, finally, in a position to explain exactly how '0' and '1' values are sent via QKD pulses between Alice and Bob. A few paragraphs back, we mentioned that this central peak emerged from the combination of double pulse assuming that Alice's and Bob's interferometers were aligned more or less

precisely. This is in fact where the (basis, value) modulation enters the picture. First, we need a few basic facts from optics[2]:

- When a light ray is incident on a surface and the material on the other side of the surface has a higher index of refraction (i.e. a lower speed of light than the medium that the light is travelling in), then the reflected light ray is shifted in its phase by exactly one half a wavelength.
- When a light ray is incident on a surface and the material on the other side of the surface has a lower index of refraction, the reflected light ray does not have its phase changed.
- When a light ray goes from one medium into another, its direction changes due to refraction but no phase change occurs at the surfaces of the two mediums.
- When a light ray travels through a medium, such as a glass plate, its phase will be shifted by an amount that depends on the index of refraction of the medium and the path length of the light ray through the medium.

With these facts as background, we can see how '1' and '0' values work in a phase-shifted QKD system. Figure 10-6 provides a schematic overview of how the system works.

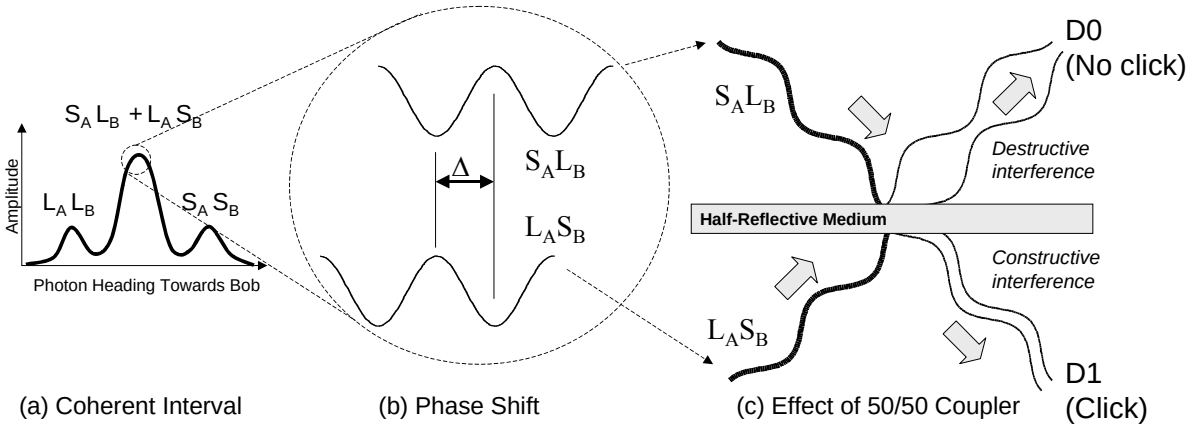


Figure 10-6. Signaling '0' and '1' Values via Phase-Shifting of Mach-Zehnder Interferometers.

As depicted, the central peak within a photon pulse contains a coherent interval (a) during which two distinct wave paths are present simultaneously. A close-up view of these waves, as in (b), shows that in general the two distinct waves have different phases – that is, the phase of the wave as it traveled through the $S_A L_B$ path is offset by some phase shift, Δ, from that which traveled through the $L_A S_B$ path. Finally, at the right, we see how these two waves interact with the final 50/50 coupler in the system to present constructive interference for one detector (click) but destructive interference for the other (no click).

Thus Alice can signal '0' and '1' values to Bob merely by adjusting the relative phases of these two waves, i.e, by adjusting the phase delta value (Δ) on a per-pulse basis. Alice does this by setting her phase shifter accordingly for each transmitted pulse.

The phase-encoded variant of BB84 works as follows. Alice encodes the 0 or 1 value for a single photon in either of two randomly selected non-orthogonal bases. She represents the 0 value by either the phase shift of 0 (basis 0) or $\pi/2$ (basis 1), and represents the 1 value by either π (basis 0) and $3\pi/2$ (basis 1).

[2] These bulleted items excerpted from a web page by David M. Harrison, University of Toronto.

Thus Alice can apply one of four phase shifts (0, $\pi/2$, π, $3\pi/2$) to encode four different corresponding (basis, value) pairs of the key as (00, 01, 10, 11). This is achieved by applying four different voltages to the phase shifter on the transmitter side. Let's assume that the voltages are 0 V for the phase shift of zero radians, 2 V for $\pi/2$, 4 V for π and 6 V for $3\pi/2$. It can be seen that the voltage on the phase shifter can be derived as a sum of basis and value bits via a summing amplifier.

When their phase difference is equal to 0 or π, Alice and Bob are using compatible bases and obtain nominally identical results (assuming a number of unrealistic assumptions such as zero noise, no photon loss, etc.). In such cases, Alice can infer from the phase shift she applied the detector chosen by the photon at the Bob's end, and hence the bit value Bob registered. By the same process of logic, Bob can deduce which value Alice transmitted. However, when Alice and Bob didn't randomly agree on the same basis (i.e. when their phase difference equals $\pi/2$ or $3\pi/2$), the bases are incompatible and the photon chooses randomly which APD it goes to. This is summarized in the table below.

Alice		Bob	
Tx Value	Basis	Basis	Rx Value
0	0	0	0
0	0	1	?
0	1	0	?
0	1	1	0
1	0	0	1
1	0	1	?
1	1	0	?
1	1	1	1

Alice			Bob			
ϕ_A	Value, Basis	Voltage	ϕ_B	$\phi_A-\phi_B$	Compatible?	Det0, Det1
0	00	0	0	0	yes	10
0	00	0	$\pi/2$	$3\pi/2$	no	?
$\pi/2$	01	2	0	$\pi/2$	no	?
$\pi/2$	01	2	$\pi/2$	0	yes	10
π	10	4	0	π	yes	01
π	10	4	$\pi/2$	$\pi/2$	no	?
$3\pi/2$	11	6	0	$3\pi/2$	no	?
$3\pi/2$	11	6	$\pi/2$	π	yes	01

10.2 Opto-Electronic Subsystem for the Mark 2 Weak-Coherent QKD Link

Figure 10-7 presents a complete schematic of our full weak-coherent QKD link, including the transmitter at Alice, the receiver at Bob, and the standard telecom fiber between Alice and Bob.

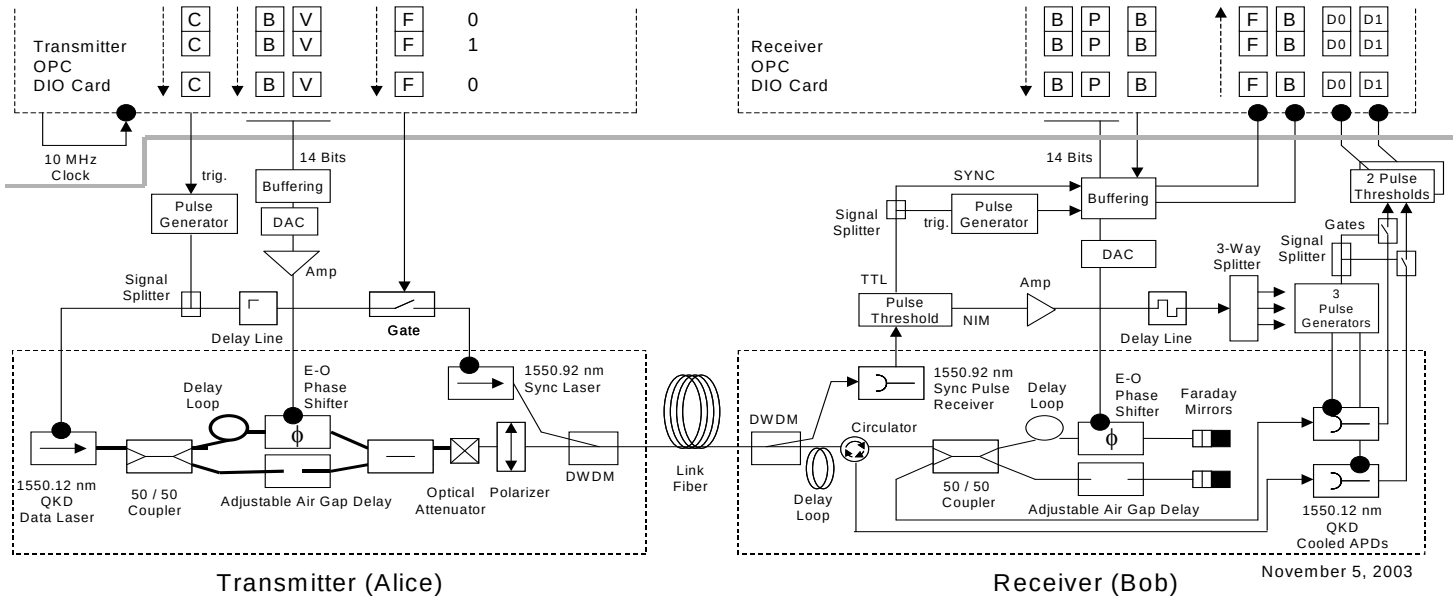


Figure 10-7. Weak-Coherent Photonics Link for Phase-Encoded BB84 at 1550 nm.

Alice provides the clock source C for both transmitter and receiver. All clocking in this system ultimately derives from a single trigger supplied from higher layers in the protocol stack, and drawn in Figure 10-7 as descending from above the Transmitter suite. As we shall see in subsequent sections, this clock comes from the Digital I/O (DIO) card on the Optical Process Control computer.

The rising edge of this signal serves as a trigger for the transmitter suite. It triggers a pulse generator whose output is split into two pulses: one drives the 1550.12 nm QKD data laser to create data pulses and the other the drives the 1550.92 nm sync laser through a gate and delay line. Framing information is encoded on the clock pulse by using the gate to produce a missing pulse in response to a '0' on the [F] line. The delay line provides a stable time relationship between the data and synch pulses, and is chosen so that the sync pulse is transmitted about 20 ns after its associated data pulse.

The data pulse passes through the unequal-path Mach-Zehnder interferometer where one arm applies phase shift modulation to the pulse. A D-to-A converter drives an electro-optic modulator with an analog voltage that produces the Basis and Value phase shifts clocked from the Transmitter OPC DIO card. In the other arm a manually adjustable air gap delay line allows fine-tuning of the interferometer differential delay. After exiting the interferometer the data pulse is attenuated to a power level that represents, on average, 0.1 to 1.0 photons per pulse. A polarizer then removes mistimed replicas of the data pulse that may have been generated by misaligned polarization-maintaining components in the interferometer. At the Transmitter output the data pulse is combined with the sync pulse in a DWDM optical multiplexing filter. The pulses are timed such that the two data pulses are separated by 17.8 ns and the sync pulse lags the second data pulse by 20.5 ns.

At the receiver the sync and data pulses are separated with a DWDM filter and the sync pulse is detected with a PIN-FET receiver. This signal is shaped in a pulse thresholding circuit that produces two outputs: a 100 ns TTL-level clock signal that is sent to the Receiver OPC DIO card and a 4 ns NIM-level APD gate-timing pulse that triggers the APD gate-pulse generators and the pulse generator driving the gates in the APD output lines. The output line gates are timed to pass only the demodulated data signal from the APDs and block noise due to spurious pulse reflections. An adjustable delay line in the NIM pulse interconnection allows fine-tuning of the APD gate-pulse timing.

The data pulse passes through a fiber delay loop to adjust its timing with respect to the sync pulse and then through a circulator that is the input to the interferometer demodulation circuit. This interferometer is a folded version of the Mark 1 design and is independent of the input polarization to accommodate the uncontrolled incident polarization at the receiver. Faraday mirrors at the ends of the unequal-length arms reflect light in such a way that the polarization of the light returning to the beam splitter is the same for each arm, producing interference with high visibility[3]. A Basis value [B] is clocked out of the Receiver OPC DIO board and applied to the electro-optic modulator through a D-to-A converter to produce a phase shift of either 0 or $\pi/2$. A pair of cooled APDs, biased above avalanche breakdown only during the time a data photon is expected to arrive, detect the interferometer outputs, one from the beam splitter and the

[3] Kersey, A., Marrone, M. and Davis, M., "Polarization-Insensitive Fiber Optic Michelson Interferometer," *Electron. Lett.*, 1991, **27**, p. 518.

other from the circulator. After gating to select only the data pulse, the APD signals are shaped by threshold detectors and passed as "0" or "1" to the Receiver OPC DIO card [D0] or [D1] channel.

A phase-correcting feedback signal, derived by the Receiver from training frames sent by the Transmitter (see Section 10.8), is used to maintain phase stability between the Transmitter and Receiver interferometers as path lengths change with temperature and stress. This phase-correcting signal is applied to the Receiver interferometer electro-optic modulator through the [P] channel of the Transmitter OPC DIO card. Phase correction is also necessary when a Transmitter and Receiver first connect during a start-up or switching operation to obtain the phase-matched condition needed for

Photonic Device	Description
Polarization-Maintaining Fiber	A single strand of commercial fiber that maintains optical polarization. This type of fiber is not used for transmission in today's telecommunications infrastructure and hence we use it only *within* a transmitter or receiver, but do not expect it to be present *between* Alice and Bob.
Non-Polarization Maintaining Fiber	A single strand of conventional single-mode (SM) fiber as widely deployed in today's telecommunications infrastructure. This type of fiber does not preserve optical polarization.
1550.12 nm QKD Data Laser	Standard telecommunications laser operating at a wavelength of 1550.12 nm. This laser is pulsed for a very short interval, then highly attenuated, so that the resulting pulse will usually consist of 0 photons, sometimes 1 photon, but rarely more than 1 photon. This resulting "single" photon is then phase-encoded with Alice's qubit in a random basis.
Variable Optical Attenuator	An attenuator that may be adjusted in order to greatly reduce the number of photons that pass from out of the transmitter. This attenuator is adjusted until it very rarely passes more than a single photon from one laser pulse.
50/50 Coupler	A component that splits light equally into two output fibers from either of two input fibers.
Delay Loop	A length of polarization-maintaining fiber with a specified propagation delay.
E-O Phase Shifter	An electro-optic waveguide device whose refractive index changes with applied voltage, thus shifting the phase of light that passes through it.
Adjustable Air Gap Delay	A variable optical delay obtained by adjusting the length of an open-air optical path.
Optical Power Monitor	Debugging tool. High sensitivity optical power meter.
DWDM (1550.12-1550.92)	Dense Wavelength Division Multiplexer. Used to multiplex together the sync pulse at 1550.92 nm with the QKD data pulse at 1550.12 nm at the transmitter, and to demultiplex the same wavelengths at the receiver. This allows the bright and QKD pulses to share one optical fiber.
1550.92 nm Sync Source	A DFB diode laser operating at 1550.92 nm and modulated through its drive current.
Link Fiber	Standard Corning SMF-28 single mode, non-polarization maintaining fiber.
1550.92 nm Sync Pulse Receiver	Integrated InGaAs photodiode and GaAs amplifier designed for telecommunications use.

1550.12 nm QKD Cooled APDs	Sensitive InGaAs avalanche photodiodes (APDs) that have been thermo-electrically cooled to lower their dark current (i.e. detector noise). These detectors are also gated, i.e., a bias voltage is applied for a very short window around the expected arrival time of the QKD photon at 1550.12 nm. This helps further increase detector sensitivity and reduce noise.
Polarizer	A fiber-coupled polarizing element with PM fiber input and SM fiber output, which passes light propagating only along the slow axis of the PM fiber.

10.3 Stabilization Issues in the Opto-Electronics Hardware

The following issues are discussed in the appropriate Photonics Subsystem document:

- Laser temperature control
- Detector temperature control
- Interferometer path lengths

10.4 The Optical Process Control (OPC) Subsystem

The Optical Process Control (OPC) subsystem is a dedicated computer with special-purpose software, designed to manage the opto-electronics hardware in a transmit or receive suite. One such OPC system is physically located at Alice and another at Bob. It has the following responsibilities:

- Perform soft-realtime housekeeping tasks for the opto-electronics hardware suite in a transmitter or a receiver (such temperature adjustments for the lasers at 10 Hz, etc.)
- Perform hard-realtime data path operations for the opto-electronics hardware suite in a transmitter or a receiver (sending and receiving qubits at 5 MHz.)
- Impose a frame structure on the weak-coherent QKD link, and establish or re-establish frame synchronization as required
- Deliver transmitted or received Raw Qframes upwards to the QKD Protocols via the interface defined by the VPN / OPC Interface Control Document.

The OPC hardware is a personal computer, running the Windows 2000 Professional operating system. It is currently implemented as an Intel-based workstation with a Pentium 4 processor, 1.5GHz, and 512 Mbytes of RAM. In addition, it contains the following interface cards:

- One Digital I/O (DIO) card, for data path operations on the opto-electronic hardware, i.e., performing all actions necessary to transmit or receive qubits at a 5 MHz rate.
- One 100 Mbps Ethernet card, for communications with the attached VPN computer.

The OPC software consists of the following major functional units:

- Tailored Matlab programs and control modules for “housekeeping” (laboratory instrument control) of the opto-electronics hardware.

- A module that sends blocks of bytes to/from the DIO card, at speeds of 5 or 10 million samples per second, in order to perform the data path operations.

- A module that implements the OPC portion of the VPN / OPC Interface.

10.5 OPC Data Path Interfaces to Opto-Electronics

This section describes the data paths between the realtime software (running on the OPC computer) and the opto-electronic hardware used in implementing the Weak-Coherent Link. By “data paths,” we mean the paths by which the individual photon (basis,value) pairs are *set* at the Source side, and *sampled* at the Detector side. These paths are entirely separate from the control paths described in the previous section, which provide a mechanism by which the control software can monitor and adjust the opto-electronic setup on an as-needed basis.

Figure 10-8 provides a simplified block diagram of these data paths. The key hardware item in this diagram is the DIO interface card present in the Source OPC and also the Detector OPC. This card provides a data path that is fast enough to clock out, or clock in, data bits at a steady 10 MHz in order to run the Weak-Coherent optical link at its design speed. In addition, it contains sufficient internal buffer memory so that it can act as a First-In, First-Out (FIFO) queue between the roughly 5 MHz optical channel and the asynchronous (as much as 132 Megabytes/second but bursty) PCI bus that connects the card to the main CPU / RAM in the OPC computer.

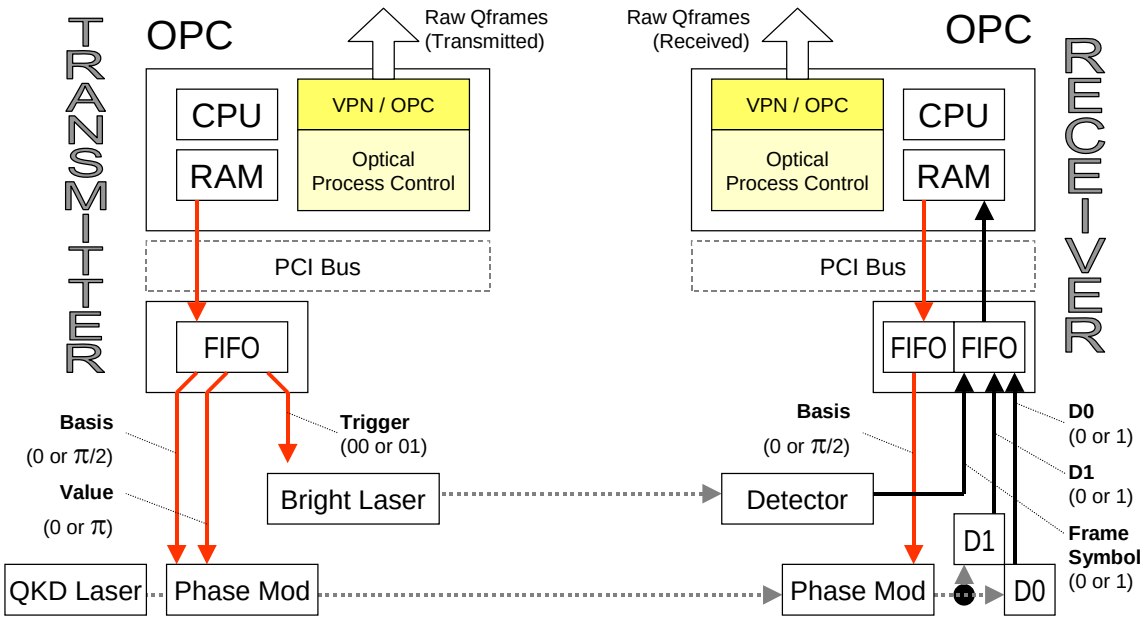


Figure 10-8. Data Flow through OPC Computer and Weak-Coherent Link.

To a first approximation, the OPC data path drives transmitter at 5 MHz so that five million QKD photon pulses are produced and modulated every second from Alice[4]. Each of these pulses is preceded, by a short and deterministic time interval implemented in the opto-electronic hardware suite, by a bright "annunciator" pulse on a different wavelength. At the receiver, these annunciator pulses drive the receiver clock so that every time such a bright pulse is received, it will trigger the sampling of the two QKD detectors and hence reception of the qubit value. Note that the same bright pulse also triggers the setting of the phase modulator at Bob so that it is ready for the next incoming QKD photon.

Conceptually, the Source OPC sends out two bits of information for every transmitted qubit (basis, value) along with a bright-pulse signal that indicates whether or not a bright pulse (on the 1550.92 nm laser source) should be transmitted along with the qubit. Thus three distinct types of information flow from the Source-side's DIO to its opto-electronic equipment for every qubit.

The Detector side works somewhat differently. Here the Detector DIO card must *set* the phase modulator (basis) at each qubit time, but must also sample "hit" values from each of the two gated detectors, D0 and D1, for each qubit. Thus at every clock step in the Detector DIO, one value flows out from the DIO while two values flow in.

Note that the transmit and receive OPCs are not at all symmetric. The transmit OPC clocks *out* three different kinds of information for every pulse: the basis for the next dim pulse, the value for the next dim pulse, and a trigger for when a pulse should be produced. The receive OPC clocks *out* a basis at every received bright pulse, and clocks *in* three other values: a received-symbol framing bit as encoded on the bright pulses, the sampled value of Detector 0 for QKD photons, and the sampled value of Detector 1 for QKD photons.

Note further that Figure 10-8 depicts the transmit-side basis as one of two phases (0 or $\pi/2$) and the value as a different phase (0 or π). As explained in Section 10.1, this is how qubits are phase-encoded in the interferometric version of the BB84 protocol. In actual implementation, the OPC's DIO card presents 14 distinct output lines, carrying the TTL voltages that correspond to a 14-bit digital value that represents the Phase Modulator voltage for a given (basis, value) setting. A Digital / Analog Card (DAC) within the opto-electronic hardware suite then converts these TTL voltages to the appropriate voltage level for setting the Electro-Optical Phase Shifter to one of four phase modulations (0, $\pi/2$, π, 3 x $\pi/2$).

Item	Description
Raw Qframes	A Raw Qframe is a fixed-size block of symbols transmitted from an OPC to its VPN. At the transmit side, it contains an indication of the bases and values that were prepared for each outbound single-photon qubit. At the receive side, it contains the basis and detector-hit values for each inbound

[4] We say that this is "to a first approximation" the way in which the system works, because in fact the transmit FIFO card runs at double speed in order to provide the right kinds of clocking information for the transmit hardware suite and to implement frame symbols in the bright pulses. This subtlety in the design is explained at some length in the following sections. In fact, though, the transmit lasers pulse at a speed that can vary somewhat (as they encode interframe symbols and frame sequence numbers), but which pulse five million times per second during most of their operation (when actually sending qubits).

	qubit. The “VPN / OPC” Interface Control Document defines the exact format of Raw Qframes.
CPU	Main CPU for the OPC computer. Pentium 4 processor, 1.5GHz.
RAM	Main memory for the OPC computer. 512 Mbytes.
DIO	National Instruments, PCI-6534.
1550.12 nm QKD Source	Source of dim (“single photon”) QKD pulses.
Phase Mod.	Electro-Optical Phase Shifter.
1550.92 nm Bright Source	Source of bright timing pulses.
Detector	1550.92 nm Bright Pulse Detector.
D0	One of Bob’s two 1550.12 nm QKD Cooled APDs.
D1	The other of Bob’s two 1550.12 nm QKD Cooled APDs.

Figure 10-9 provides a more detailed exposition of the same data paths, indicating the binary values stored within the OPC’s DIO cards at both the transmit and receive sides, and the means by which these values interface with the opto-electronic hardware suites for the weak-coherent link.

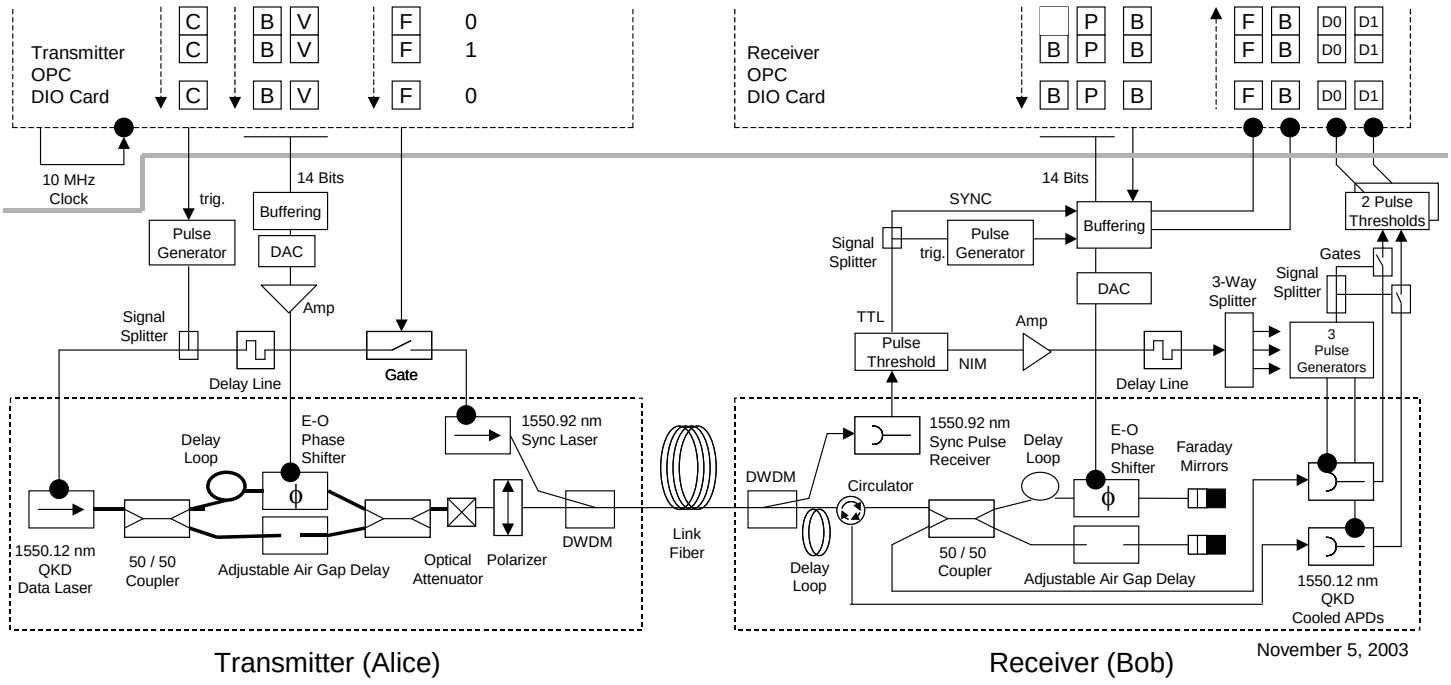


Figure 10-9. Data Path from Alice’s OPC to Bob’s OPC.

The binary values stored in the DIO cards have the following definitions:

B-V	14 Bits of phase voltage modulating Alice's (basic, value) pairs
B-P	14 Bits of phase voltage modulating Bob's basis and Path Length Control
B	Basis for a qubit
F	Framing bit; trigger (at rising edge) for generating a photon pulse
C	Transmit clock
D0	1 iff Detector 0 indicated a "hit"
D1	1 iff Detector 1 indicated a "hit"

The Frame (F) bit is employed when sending frame symbols across the bright pulse channel; its use is described in detail in a subsequent section. For now, we merely note that the transmit-side clock is derived from a 10 MHz clock running internally within Alice's OPC DIO card. A series of F values that alternate between 0 and 1 (e.g. 0101010. . .) down-convert this 10 MHz clock to triggers for the opto-electronic equipment at a 5 MHz rate, i.e., by the rising edges of the 1's.

10.6 The Relationship between Raw Qframes and Photons

This section describes the mapping between QKD information as stored and manipulated within OPC and VPN computers (Raw Qframes) and the equivalent information as it flows "across the wire" in single photons using phase-encoded BB84.

Figure 10-10 presents this mapping in graphic form. At the top, it shows a Raw Qframe as fabricated at the transmit (Alice) side. At the bottom, it shows the receiver's version of the same Raw Qframe. Each location in the Raw Qframe is numbered (e.g. #1, #2, . . .) and contains all necessary information for a single qubit. At the transmit side, this includes the basis and the value; note that the single-bit value is actually encoded in two bits, so that the transmit version of the Raw Qframe mirrors the received version.

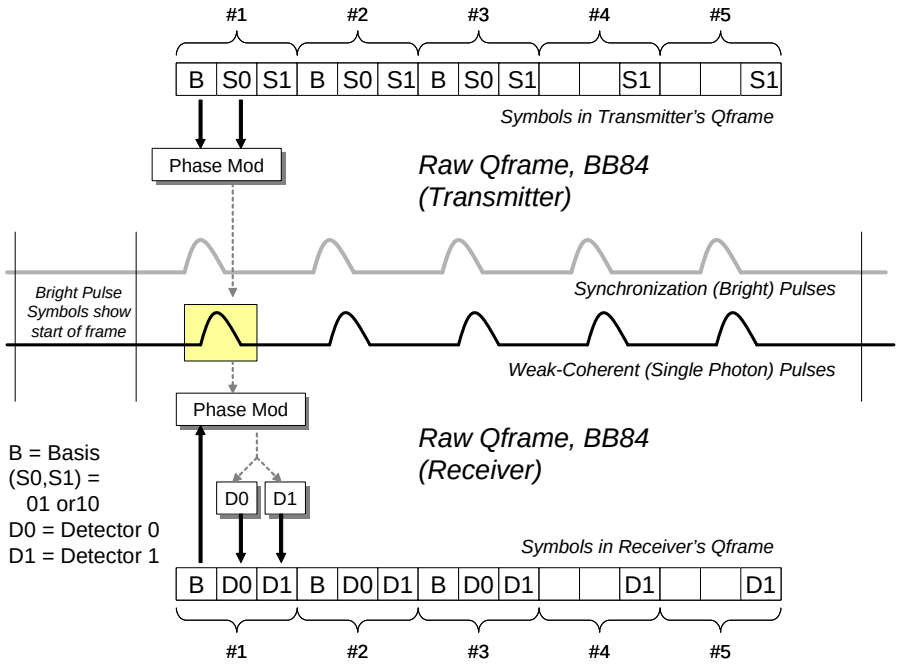


Figure 10-10. Relationship of Transmit / Receive Frames and Physical Transmission on Link.

The middle portion of Figure 10-10 sketches how these binary values are physically represented on the weak-coherent QKD channel. As shown, Alice's basis and value are combined as inputs to the transmitter's Electro-Optical phase modulator, which is used to modulate the single photons at 1550.12 nm. At the receive side, Bob must randomly choose a basis for setting the receiver, then sample for "hits" on both QKD detectors D0 and D1. These three values (B, D0, D1) must be treated as a triplet, i.e., Bob must correctly remember which value of B it employed when getting the hits on D0 and D1.

In the very simplest case, where all bits are perfectly conveyed between Alice and Bob and furthermore both Alice and Bob miraculously choose the same basis for each qubit, the transmit and receive Raw Qframes will be identical. In cases where Alice and Bob randomly select the bases, then the qubit positions (#1, etc.) will be identical for those qubits in which Alice and Bob chose the same basis, but Bob's Raw Qframe values will be meaningless for those positions in which they chose different bases. In the presence of noise or eavesdropping, Alice and Bob may have different values in even those qubit positions where they chose the same basis. All such issues will be arbitrated by the QKD Protocols running at a higher layer in the protocol stack, in the VPN computers, on the basis of the Raw Qframes given to them by the OPC entities. Thus the OPC software does not need to worry about any of these issues. It merely accumulates the Raw Qframes and hands them off.

Finally, note that the QKD link's physical channel is organized into frames rather than being merely a continuous stream of qubits. The next section describes exactly how the OPC imposes this frame structure on the QKD link.

10.7 Framing via Bright Pulses on the Mark 2 Weak-Coherent QKD Link

This section describes how the OPC imposes a frame structure on the Weak-Coherent QKD link, i.e., organizes the stream of single photons into numbered frames as required by the higher-level QKD protocols.

Figure 10-11 displays the basic mechanism for encoding two symbols (0 or 1) into the bright pulses delivered at 1300 nm. As shown, this scheme varies the timing of bright pulses to encode symbols. Thus a '1' symbol is encoded by a rising edge that is produced within some predetermined "beat" interval, while a '0' is encoded by a longer delay before that rising edge. Note that as a side-effect of this scheme, 1's occupy less time on the channel than do 0's, and hence 1's can be sent at a faster rather than 0's can be.

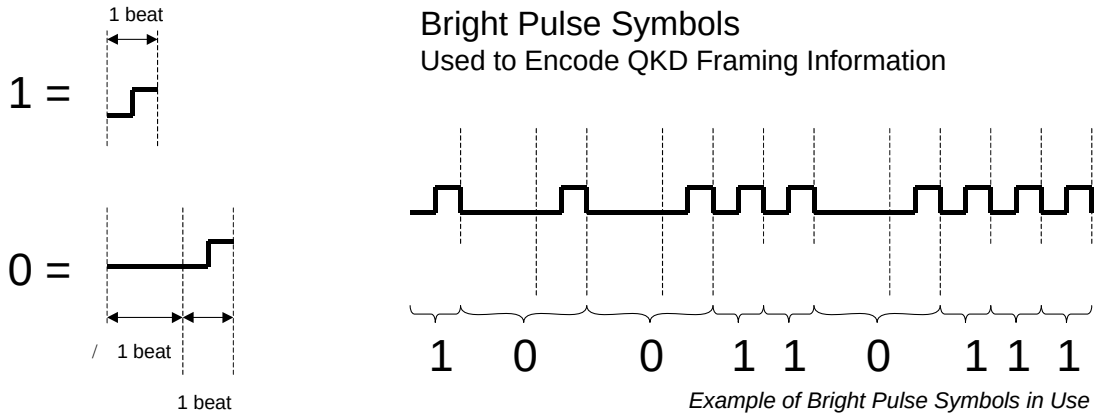


Figure 10-11. Bright Pulse Symbols as used to Encode QKD Framing Information.

Note further that a '0' symbol can be of indeterminate length, so long as its initial delay is greater than one beat. For example, even a period of a millisecond or more of dead time, followed by a rising edge, will be treated as a single '0' symbol.

These symbols are generated by the sequence of T (trigger) values clocked out from Alice's OPC by the 10 MHz internal clock. Thus a series of T values such as 01010101. . . will generate a steady stream of '1' symbols on the bright pulse channel since each rising edge of a T=1 occurs within one beat at a 5 MHz rate. However a series of T values such as 00010001. . . generates a steady stream of '0' symbols since the rising edges occur more than one beat apart.

By this mechanism – manipulating the series of T values clocked out at a steady 10 MHz – the transmitter OPC can impose a series of binary symbols on the bright pulse channel. The receiver OPC can then sample these symbols as received, via its own F channel, thus receiving the series of 0s and 1s that the transmitter encoded in its bright pulses.

Figure 10-12 shows how this additional "bright pulse channel" is used to define a frame structure for the QKD link. As depicted, the interframe mark is defined as special sequence of four or more '0' symbols followed by a '1' symbol. This interframe mark is immediately followed by 8 bits of frame type, then 32 bits of frame sequence number encoded in binary with the same '0' and '1' symbols. These 40 symbols are then followed by a fixed number of '1' symbols, each of which acts as an annuciator pulse for a QKD photon that arrives shortly afterwards.

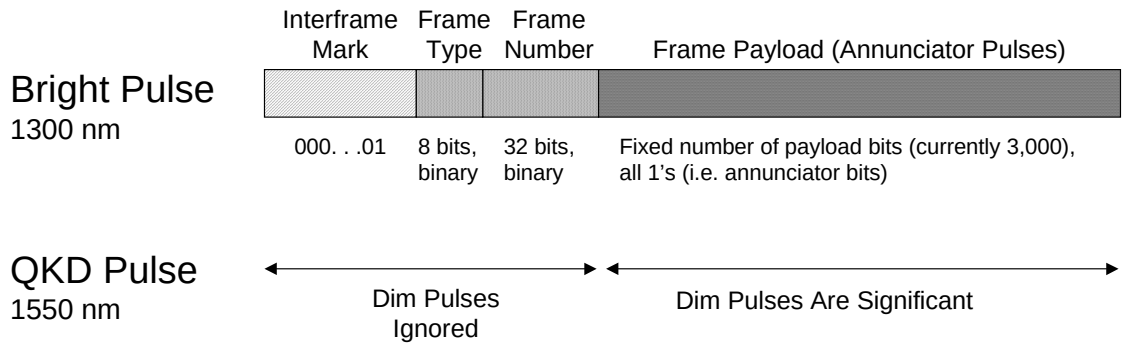


Figure 10-12. Mark 2 Weak-Coherent QKD Frame Format.

By definition, no qubit information is sent during the interframe mark. In fact, the opto-electronic hardware does not have any "state" regarding frames, so it will generate both 1550 nm and 1300 nm pulses at each trigger event from the OPC. However the settings of the transmit Electro-Optical Phase Shifter are undefined during the interframe mark and frame number (the OPC may set them to any value that is convenient) and hence the 1550 nm pulses produced during these times do not convey any qubit information.

Finally, note that the QKD protocols in Alice's VPN computer are responsible for setting the initial value for the transmitted frame sequence number. For instance, they will instruct the OPC to use a frame sequence number equal to 1 when a link is first brought into operation. Then Alice's OPC is responsible for automatically incremented this sequence number for each new frame that it creates and transmits. It is necessary for the QKD protocols to be able to set these initial frame sequence numbers so that the QKD protocols can recover in cases of crash and restart of the QKD Protocol entity on one side of the QKD link.

10.8 Data Frames and Training Frames

In normal operation, the Weak Coherent QKD link transports a mixture of data frames and training frames. Data frames convey modulated weak pulses for quantum cryptography. Training frames convey modulated weak pulses for link control, and in particular for continuous path length control. These two types of frames are distinguished by the values in their Frame Type fields; please see the Photonic Subsystem Document for more details.

Figure 10-13 shows, in high level form, how these two types of frames are intermixed on the Weak Coherent QKD link. At present, we intend to send a repetitive pattern of N data frames followed by M training frames, where the numbers of frames N and M will be determined by analysis and experimentation. This transmission plan is governed entirely by Alice; Bob has no knowledge of the plan and simply works with whatever it receives. Later we may evolve to more complex schemes in which N and M are dynamically determined based on observed link characteristics. A more detailed level of design will determine whether data frames and training frames have the same, or different, payload sizes.

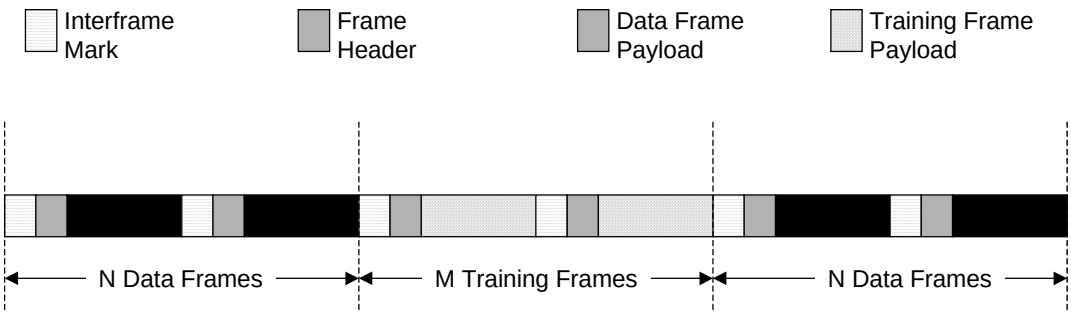


Figure 10-13. Normal Operation Alternates Sequences of Data Frames and Training Frames.

Figure 10-14 depicts two different approaches to the training frames' payloads, i.e., to the dim pulses conveyed within the body of a training frame[5]. In both cases the payload consists of modulated dim pulses as in data frames. In the simpler (a) variant, Alice uses a fixed pattern when modulating the pulses in a training frame payload. In the more complex (b) variant, Alice uses a deterministic but pseudo-random pattern when modulating the pulses, with the frame number as the seed for the pseudo-random number generator of the (basis, value) pairs.

We expect to employ the simpler (a) variant first, but may then progress to the more complex (b) variant once we have the path-control algorithms working properly.

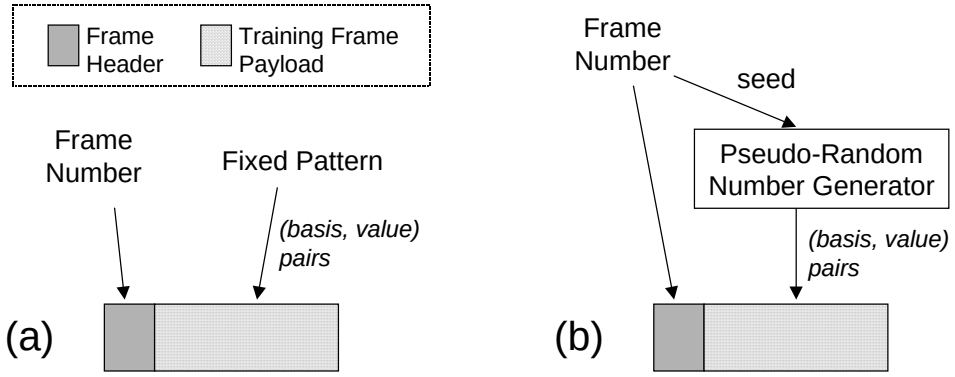


Figure 10-14. Two Different Approaches to Training Frame Payloads.

The simpler (a) variant works as follows. For each training frame, Alice prepares a well-known sequence of (basis, value) pairs that are identical for every training frame. For example, this sequence might be a repetition of the following pairs: (0,0), (0,1), (1,0), (1,1). However it is more likely that the sequence will be more complex, e.g., a series of all possible permutations of 4-qubit sequences in order to exercise unusual cases in detector behavior such as detector detection probabilities that vary depending on recent history of activity.

[5] A third approach discards the notion of training frames altogether, and instead sacrifices a select subset of bits within the data frames for training purposes. These bits may be selected by a pseudo-random algorithm in order to make it hard for Eve to determine how much training is being performed or to influence the training. We expect to explore this approach later in the design process after these simpler approaches are operational.

The more complex (b) variant works as follows. For each training frame, Alice first prepares a frame number and includes that in the frame header. Alice also uses this frame number as a seed to a deterministic algorithm such as a pseudo-random number generator, which in turn generates a sequence of (basis, value) pairs from this original seed. Alice then modulates the training frame's dim payload pulses using this derived sequence. After Bob has received the frame, Bob can extract the frame number and run the same deterministic algorithm in order to determine the sequence of (basis, value) pairs that Alice sent within this frame.

In either case, note that Bob's hardware is currently incapable of controlling the receiver's basis values "on the fly," e.g., in realtime as Bob receives a training frame. This is because there is a lengthy and essentially uncontrolled delay between when Bob's OPC CPU generates a series of basis values in its local memory, and when the corresponding sequence is clocked out from Bob's digital I/O card to control Bob's phase modulator.

As a result, Bob does not attempt to perform any special type of basis-control (phase modulation) for training frames – indeed, it treats data frames and training frames identically in this regard. However, in either the (a) or (b) variants described above, Bob can determine – without public communication with Alice – what basis values Alice used, because they are deterministic. As a result, Bob can use this knowledge to build the following table of counts as input for the Path Length Control algorithm. See the Photonic Subsystem Document for further details about this area, e.g., of how often this table is polled by the Path Length Control process, etc.

Alice (b,v)	Bob (b)	Binned Counts of Training Events Since Last Report			
		No hit	D0 hit	D1 hit	Both D0, D1 hit
(0,0)	0				
(0,0)	1				
(0,1)	0				
(0,1)	1				
(1,0)	0				
(1,0)	1				
(1,1)	0				
(1,1)	1				

10.9 The VPN / OPC Interface

This section describes the hardware and software interface between the Virtual Private Network (VPN) and Optical Process Control (OPC) computers. This interface is defined by the "VPN / OPC Interface Control Document," which should be consulted for a further level of details.

This interface provides the following functionality:

1. Define the major states of the underlying QKD link, as seen by the VPN side of the interface, and allow these states to be observed and commanded.

2. Establish, maintain, and if necessary re-establish communications between software in the VPN and OPC computers as the system starts up, daemons crash, etc. Specifically this interface provides communications between the QKD Daemon in the VPN computer and the LabView software in the OPC computer.

3. In setup, provide a means by which the VPN computer can determine capabilities of the attached OPC computer (e.g. whether it can perform the BB84 protocol, expected error rates, etc.).

4. In setup, provide a means by which the VPN computer can command the OPC computer to perform initialization and configuration of its attached electronic and optical devices.

5. In actual operation, transport a continual sequence of raw Qframes of transmitted or received symbols from the OPC sub-system to the VPN sub-system, and in particular to the QKD Daemon, so that these raw frames can be used as input to the QKD protocols (such as sifting, error correction, etc.)

10.9.1 OPC and LINK States as Seen by the VPN Side

This section defines the major states of the Optical Process Control (OPC) entity and of the QKD link that it controls, as seen from the VPN side of the interface. In general, the VPN acts as the master side of this relationship since we expect that VPN entities, but not OPC entities, will be accessible from other network nodes. Thus eventually we expect all network monitoring and control to pass through the VPN side of the interface and "reach down" to the OPC.

Figure 10-15 below defines the OPC's major states as seen from the VPN, i.e., as reported in internal state tables within the VPN. Two tables below the figure define the meaning of the OPC states and the transitions between them.

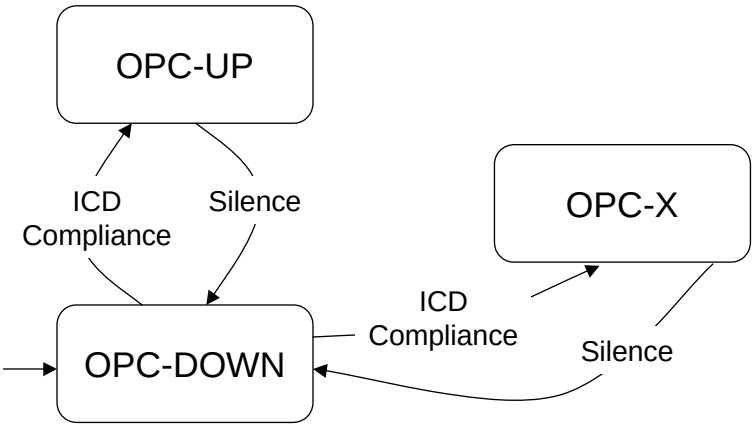


Figure 10-15. OPC States as Seen from the VPN.

OPC States	Description
OPC-DOWN	OPC is not communicating with the VPN in a manner compliant with the VPN / OPC Interface. This may be because the OPC software entity is not running, its computer is powered off, its Ethernet is broken, it is running

	code sufficiently buggy so that it cannot implement the ICD properly, etc. Upon startup, the OPC entity remain in this state (as viewed from the VPN) until it enters normal or experimental operation, as described below.
OPC-UP	The OPC is operating normally.
OPC-X	The OPC software is running and compliant with the VPN / OPC Interface, but is not operating normally. It could be running test code, alignment patterns, data-gathering routines, or whatever; but it is not attempting to function as a operational QKD link. (The intent of this state is to eventually allow testing of remote nodes, via messages relayed by the VPN.)

OPC State Transitions	Description
OPC-DOWN to UP	The OPC has completed its startup interactions with the VPN as defined in the ICD, and has indicated that it is operational (as opposed to experimental). This transition should be autonomous, rather than cued by operator command, etc., so that the system automatically restarts after power outages.
OPC-UP to DOWN	The OPC has ceased to properly implement its side of the ICD, either by sending incorrect messages or by falling silent for too long a period.
OPC-DOWN to X	The OPC has completed its startup interactions with the VPN as defined in the ICD, and has indicated that it is experimental (as opposed to operational).
OPC-X to DOWN	The OPC has ceased to properly implement its side of the ICD, either by sending incorrect messages or by falling silent for too long a period.

The QKD "LINK" states are defined only when the OPC entity is functioning normally and communicating with the VPN, i.e., in the OPC-UP state. Figure 10-16 below defines major LINK states as seen from the VPN. Two tables below the figure define the meaning of the LINK states and the transitions between them.

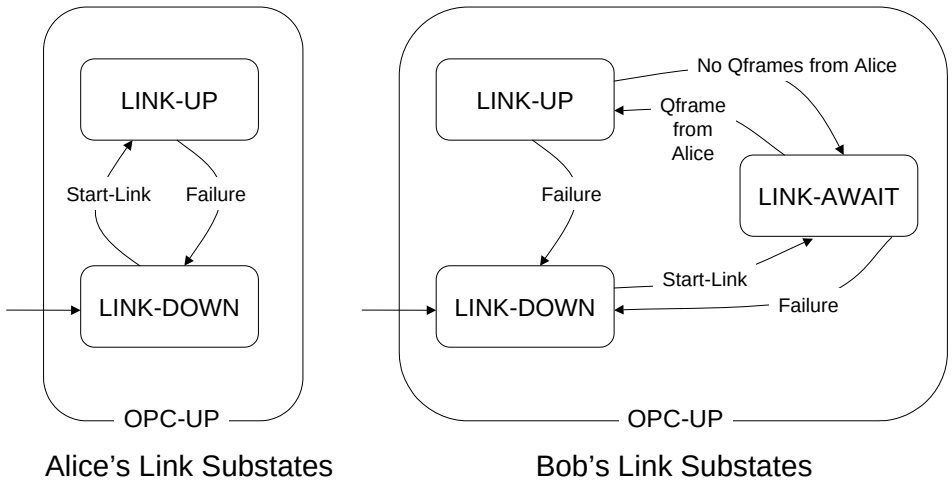


Figure 10-16. LINK States as Seen from the VPN.

LINK States	Description
LINK-DOWN	This is the initial LINK state when the OPC enters its OPC-UP state. At Alice (the source), no pulses are sent on the optical channel. At Bob (the detectors), no values are read from the opto-electronics equipment ,or equivalently, the values are read but ignored.
LINK -UP	At Alice, the OPC is running its source suite in normal operational mode and transmitting frames of modulated qubits. At Bob, frames are being received from the opto-electronic suite, are being processed within the OPC, and are being sent onwards to the VPN as Raw Qframes.
LINK -AWAIT	This state is not used in Alice. At Bob, the OPC is running its receiver suite and sampling all received impulses, but no frames are being received across the photonic link.

LINK State Transitions	Description
LINK -DOWN to LINK-UP	At Alice, the OPC performs this transition in response to an explicit Start-Link command sent from the VPN (see the ICD for details). At Bob, this transition is not allowed; instead Bob moves to the LINK-AWAIT state when bringing up its link, as described below.
LINK -UP to LINK-DOWN	The OPC makes this transition autonomously, in response to local information at the OPC such as opto-electronic equipment failures that it detects. It then reports the transition to the VPN by mechanisms defined in the ICD.
LINK -DOWN to LINK-AWAIT	At Alice, this transition is not allowed, since Alice does not have a LINK-AWAIT state. At Bob, the OPC performs this transition in response to an explicit Start-Link command sent from the VPN (see the ICD for details).
LINK -AWAIT to LINK-DOWN	At Alice, this transition is not allowed, since Alice does not have a LINK-AWAIT state. At Bob, the OPC makes this transition autonomously, in response to local information at the OPC such as opto-electronic equipment failures that it detects. It then reports the transition to the VPN by mechanisms defined in the ICD.
LINK -AWAIT to LINK-UP	At Alice, this transition is not allowed, since Alice does not have a LINK-AWAIT state. At Bob, the OPC performs this transition when it receives a correctly-formatted frame from Alice across the photonic channel. It then reports the transition to the VPN by mechanisms defined in the ICD.
LINK -UP to LINK-AWAIT	At Alice, this transition is not allowed, since Alice does not have a LINK-AWAIT state. At Bob, the OPC performs this transition when a specified interval of time passes without receipt of any correctly-formatted frames from Alice on the photonic channel. It then reports the transition to the VPN by mechanisms defined in the ICD.

10.9.2 Interface Implementation

At the hardware layer, this interface is implemented by a 100 Mbit/second Ethernet link between the VPN and OPC computers. This must be a direct connection between the two machines, with no intervening bridges or hubs. It acts as a “virtual bus” carrying unprotected UDP / IP datagrams between the two computers.

The VPN and OPC interfaces on this link must be configured so that this link cannot be accessed from any entity except those directly resident on the VPN or OPC computers, or other devices needed for development work such as the laboratory cvs server. (This privacy helps to prevent an attack in which some unauthorized entity spoofs the VPN or OPC entities in this protocol.)

At the transport layer, this protocol employs its own semi-reliable delivery mechanisms atop UDP datagrams. Please consult the ICD for a description of how this transport works, and why it is used instead of a more conventional protocol such as TCP.

Figure 10-17 shows the basic operation of this protocol at the Source suite in schematic form. As can be seen, the protocol consists of two basic phases: setup and operational. In the setup phase, the VPN computer commands the OPC computer to initialize, determines its capabilities, and then sets the desired operational parameters. The protocol then enters an operational phase, in which the OPC computer sends up a series of raw Qframes as they are transmitted. From time to time, the VPN computer calculates the average error rate, as determined from these raw Qframes and QKD protocols with its peer, and transmits this error-rate information to the OPC computer. The OPC computer may then, if desired, adjust its operation, e.g., so as to attempt to reduce the error rate.

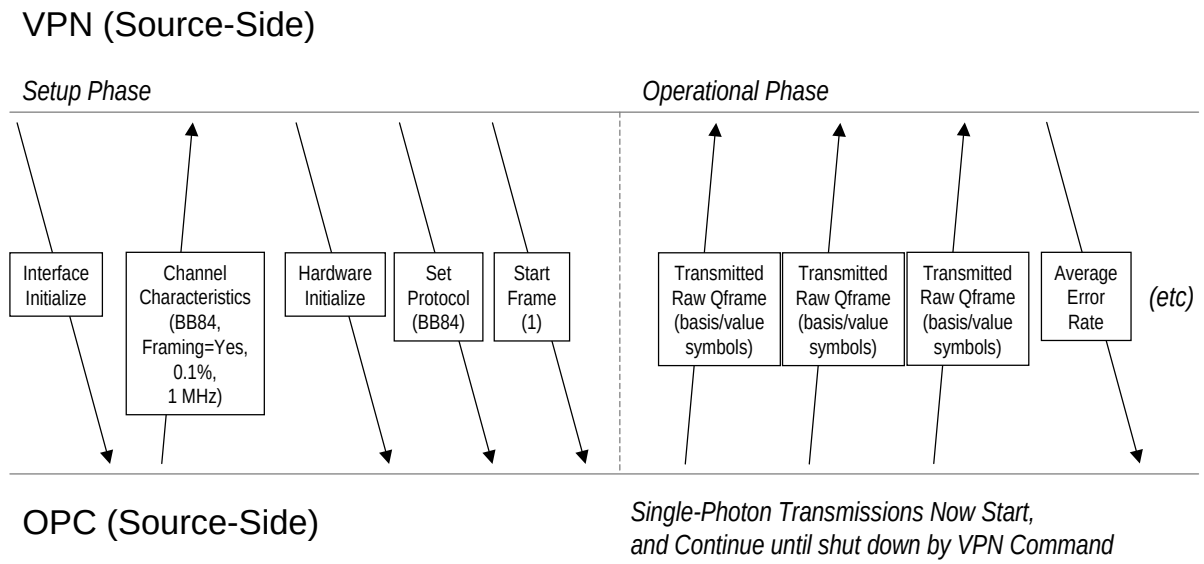


Figure 10-17. Overview of VPN / OPC Interface (Source Side).

As shown in Figure 10-18, a very similar version of the protocol is used at the Detector suite as well. The main difference here is that the Detector side does not establish the frame numbers.

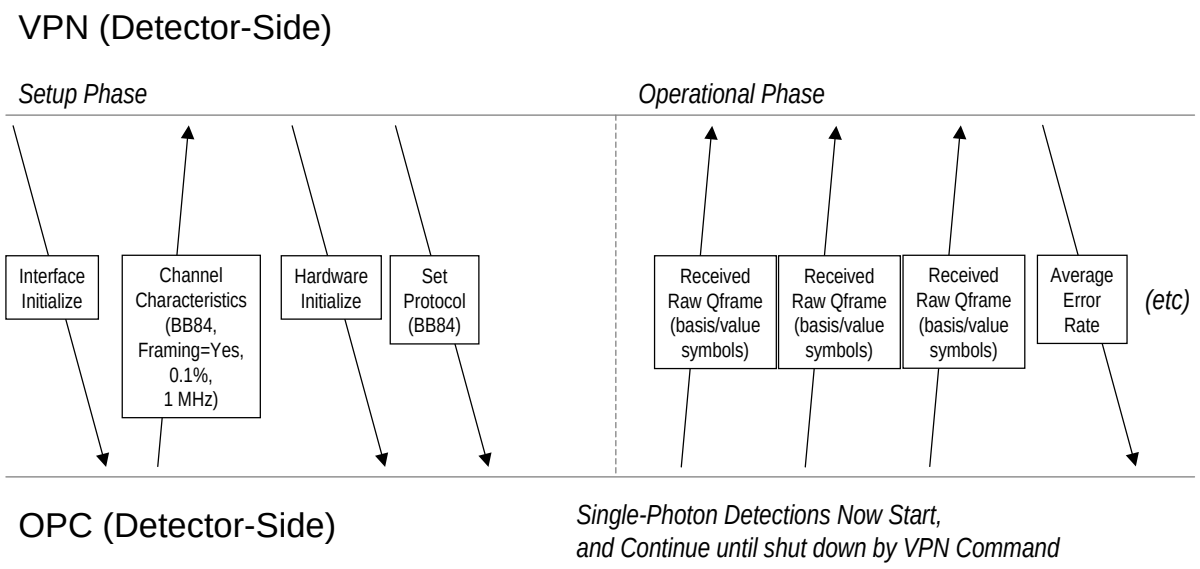


Figure 10-18. Overview of VPN / OPC Interface (Detector Side).

11 The Mark 1 Entangled Link

This section describes our Mark 1 Entangled QKD link, based on correlated polarizations of pairs of entangled photons generated by Alice. This system is designed to work through telecommunications fiber, so it contains a novel polarization control subsystem to undo the "polarization scrambling" caused by telecommunications fiber, and thus allows accurate reception of polarized QKD photons at Bob. At upper layers, it employs a variant of the BB84 protocol (rather than, say, the Ekert protocol).

Our Mark 1 Entangled Link implementation was strongly influenced by the work of Prof. Nicolas Gisin's group at the University of Geneva, although the Mark 1 Entangled Link does have some significant differences from any of the various Geneva systems reported to date. We would particularly like to acknowledge a system documented by Gregoire Ribordy et al. in a paper[6] that is exemplary in its clarity and thoroughness. This paper was a great help to us. Another very useful paper[7] by Gisin et al. discussed the decoherence problems caused by polarization mode dispersion and chromatic dispersion.

11.1 The Entangled Link in Broad Context

As the DARPA Quantum Network becomes built out, it will incorporate a number of heteregenous types of underlying QKD links. The Mark 1 Entangled link is thus only the first of several QKD links in the DARPA Quantum Network, and we expect that it will be the only link of this particular type in the overall network.

Figure 11-1 shows one categorization of the universe of possible types of QKD links. It is not intended to be a complete typology but rather to concisely depict certain relevant aspects of the Mark 1 Entangled link that set it apart from other kinds of QKD links.

			Telecom Fiber	Freespace
Weak Coherent		One-way	*Mark 2 Weak Coherent Link*	--
		Plug and Play	--	--
Entangled Photon Pairs	Polarization	Source Inside Alice	**Mark 1 Entangled Link**	--
		External Source	--	--
	Phase	Source Inside Alice	--	--
		External Source	--	--

Figure 11-1. Mark 1 Entangled Link in Context.

[6] G. Ribordy, J. Brendel, J-D. Gautier, N. Gisin, and H. Zbinden, "Long-distance entanglement-based quantum key distribution," Phys. Rev. A, v. 63, 012309 (13 December 2000).

[7] N. Gisin, J. Brendel, J-D. Gautier, B. Gisin, B. Huttner, G. Ribordy, W. Tittel, H. Zbinden, "Quantum cryptography and long distance Bell experiments: How to control decoherence," quant-ph/9901043 (15 January 1999).

See Figure 11-2 for a schematic representation of this view of the Entangled Link. Other links in the DARPA Quantum Network employ differing types of sources, detectors, and modulation.

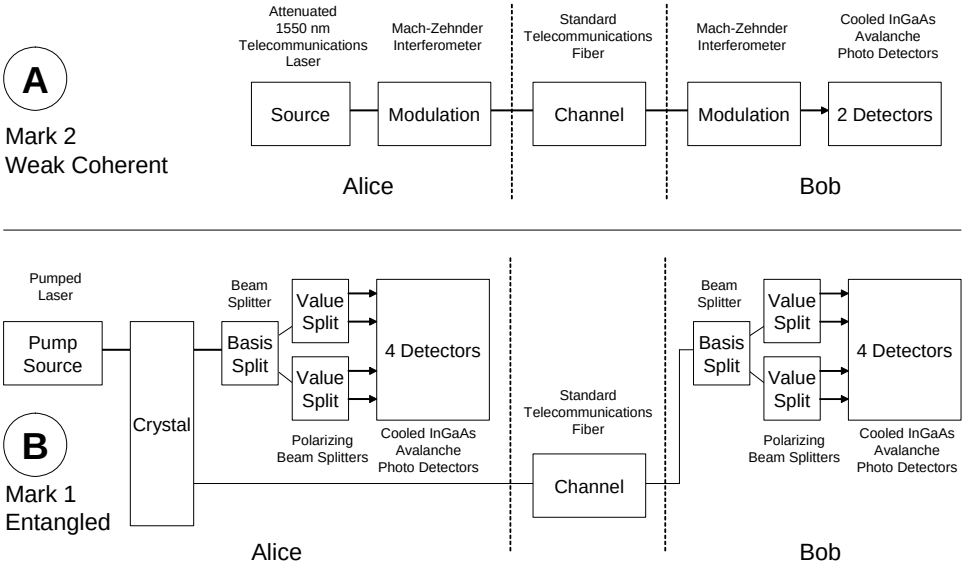


Figure 11-2. Characteristics of the Mark 1 Entangled Link

The following aspects of the Mark 1 Entangled link are fundamental to its overall operation and performance. It is important to understand these fundamental points, why they occur, and their implications on the overall system.

Entangled Pairs. The Mark 1 Entangled link is based on entangled photon pairs, because systems based on highly attenuated laser pulses (weak coherent systems) have been shown to be vulnerable to at least theoretical forms of attack from Eve, by Brassard et al in a paper[8] from 1999. Such attacks are generally termed "Photon Number Splitting" (PNS) attacks.

Telecommunications Fiber. This link is designed to run through telecommunications fiber as widely deployed today. Thus we have chosen to transmit entangled photons in the 1550 nm window for maximal distance through this fiber. At present, these photons can be best detected by certain kinds of commercial InGaAs APDs cooled to approximately –50 degrees Centrigrade. These cooled detectors form a bottleneck in the overall link performance, as they require on the order of 200 ns to recover between detection events. It is likely, however, that our first version of the entangled source will not have a particularly high rate of generating entangled pairs, and that this bottleneck will not prove important for a while. More important is the relatively low Quantum Efficiency (QE) of these APDs. Since entanglement-based cryptography requires the correlated detection of two photons, the raw (unsifted) bit rate is limited by the *square* of a single APD's detection probability.

[8] Brassard, G., Mor, T. and Sanders, B. C., "Quantum cryptography via parametric downconversion," *quant-ph/9906074*.

Polarization Modulation. The Mark 1 Entangled link uses polarization modulation to encode the (basis, value) pairs needed for quantum cryptography. Such modulations can be produced relatively easily in Alice and Bob. In fact, the random selection of the value happens in the pair generation process, and random selection of basis can be performed purely passively by interposition of a beam splitter. This simplicity stands in contrast with the relative complexity of phase modulation, which requires carefully tuned Mach-Zehnder interferometers and an external source of randomness that drives deterministic phase modulators. However, polarization is quite difficult to transmit through a telecommunications fiber, which generally acts as a "polarization scrambler." Thus an important part of the Mark 1 Entangled Link design is the polarization control at Bob.

BB84 Protocols. The Mark 1 Entangled Link runs the BB84 cryptographic protocols, rather than the Ekert protocols. We do this because we already have a well-debugged version of BBN's BB84 protocol stack, which we can directly employ on this new link. In later stages of this project, we may also implement the Ekert protocols.

Despite the fact the the Mark 1 Entangled link has such specific characteristics, the overall architecture for the DARPA Quantum Network provides an interface between the QKD link subsystem and the remainder of the overall network system, so that the rest of the system has very little knowledge of the underlying QKD link technology. This makes it easy to add new kinds of QKD links as we build out the network, and to upgrade existing links as technology improves, e.g., as single photon sources and better detectors become available.

11.2 Basic Principles of the Mark 1 Entangled Link

This section discusses the basic principles of the Mark 1 Entangled Link. It first describes how the hardware implements the BB84 protocol with entangled pairs, and then provides and overview of how the necessary optical control functions (polarization maintenance, framing) are implemented.

11.2.1 BB84 (Basis, Value) Modulation of Entangled Pairs

Figure 11-3 shows how the entangled source emits polarization-entangled photon pairs. It depicts the two emission cases that can occur: either the "extraordinary" polarization (E) photon is emitted along one path and the "ordinary" polarization (O) photon on the other, or vice versa. These photons pass a short distance in free space, then enter separate fibers through a coupling device. One device is set so that extraordinary polarization (E) photons enter the fast (F) axis of the associated fiber, and O polarized photons enter the slow (S) axis. In short, it maps E→F and O→S. The other twists the fiber by 90^o to obtain just the opposite result; on that fiber, E→S and O→F.

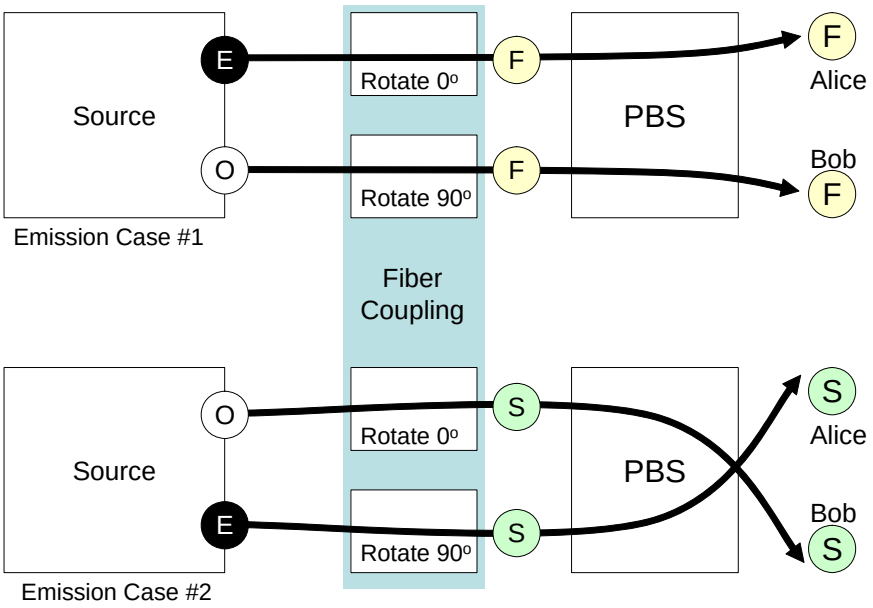


Figure 11-3. The Polarizing Beam Splitter (PBS) Produces Entanglement.

Thus in Emission Case #1, both Alice and Bob receive photons polarized along the fast axis (F) of the fiber. In Emission Case #2, both receive photons polarized along the slow axis (S). Note that it is entirely random which of the two emission cases occurs, and thus the resultant polarization sent to Alice and Bob is entirely random (but identical). As we shall see, this polarization encodes the 'value' used in BB84.

One minor detail is important enough to mention here. In order to ensure that the two photons are properly entangled, they must impinge upon the Polarizing Beam Splitter (PBS) at the same time. Thus the two Fiber Coupling apparatuses must be carefully positioned, by hand, to ensure that this is true. See Figure 11-4. The better the positioning, the higher fidelity the entanglement. These are expected to require relatively infrequent readjustment, perhaps on the order of weeks or months.

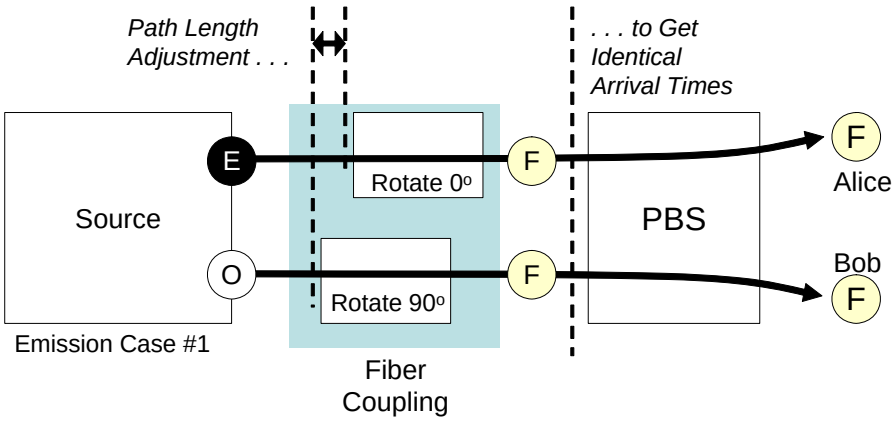


Figure 11-4. Path Length Adjustments to Obtain High-Fidelity Entanglement.

At the end of this stage, we see that identically-polarized photons are prepared for both Alice and Bob, and that the entangled polarization of these photons encodes the 'value' used in BB84. The next stage is

for Alice and Bob to independently select random bases for measurement, and then to measure these photons. This is accomplished via identical detector suites in Alice and Bob.

Figure 11-5 shows how BB84 is implemented, in highly schematic form. In case A at the top of the diagram, we see a pair of "F" encoded photons heading towards detectors at Alice and Bob. These photons first pass through a beam splitter, and randomly choose one of its output arms. This provides the random selection of the 'basis' for BB84. A subsequent polarizing beam splitter then deterministically sends the polarized photon to the appropriate detector.

This complete set of operations gives rise to a (basis, value) pair. In case A, therefore, we have labeled Alice's photon as "1F" indicating that she measured it in 'basis' 1, and detected its 'value' of F. The resultant (basis, value) pair can be represented as (1,0). At the right side of the diagram, for case A, we see that Bob makes an identical measurement and derives an identical value (0,1).

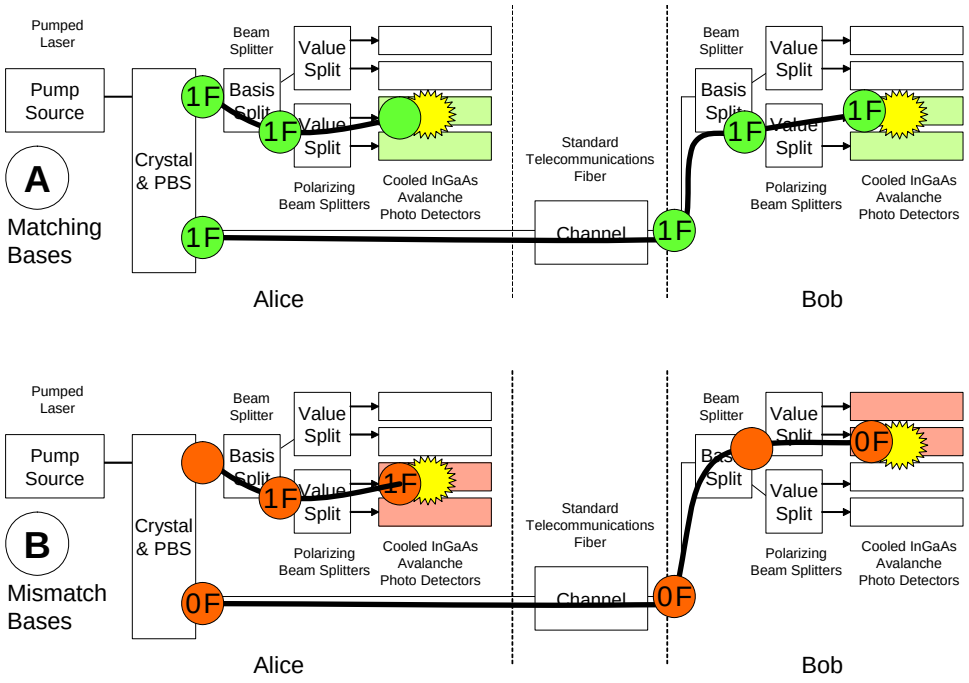


Figure 11-5. Entangled Link in Operation with (A) Matched vs. (B) Mismatched Bases.

In case B, at the bottom of Figure 11-5, we see what happens when the bases don't match. Here Alice's photon took the same path through its basis selection (beam splitter) as in case A, but Bob's took the other path. Hence Alice's and Bob's bases don't match. These mismatches will be winnowed out in the sifting phase of BB84, precisely as in its non-entangled version.

Although not shown in these high-level figures, the two detector suites do implement non-orthogonal bases just as in the original BB84. They do so by interposing a 45^{o} rotator along one of the basis paths, i.e., between the beam splitter and the polarizing beam splitter. This has exactly the same function as in the original BB84, namely, to ensure that Eve is forced into making a measurement in an unknown basis, and thus necessarily introducing noise into the system when attempting intercept-resend attacks.

Now that the basic principles of entangled BB84 have been outlined, **Figure 10-1** highlights the major features of our Mark 1 Entangled link. As shown, the transmitter at Alice contains both a source of entangled photons – i.e. a crystal pumped by a pulsed laser source – and detector suite containing 4 APDs, one for each possible (basis, value) pair. Bob contains a second, identical detector suite.

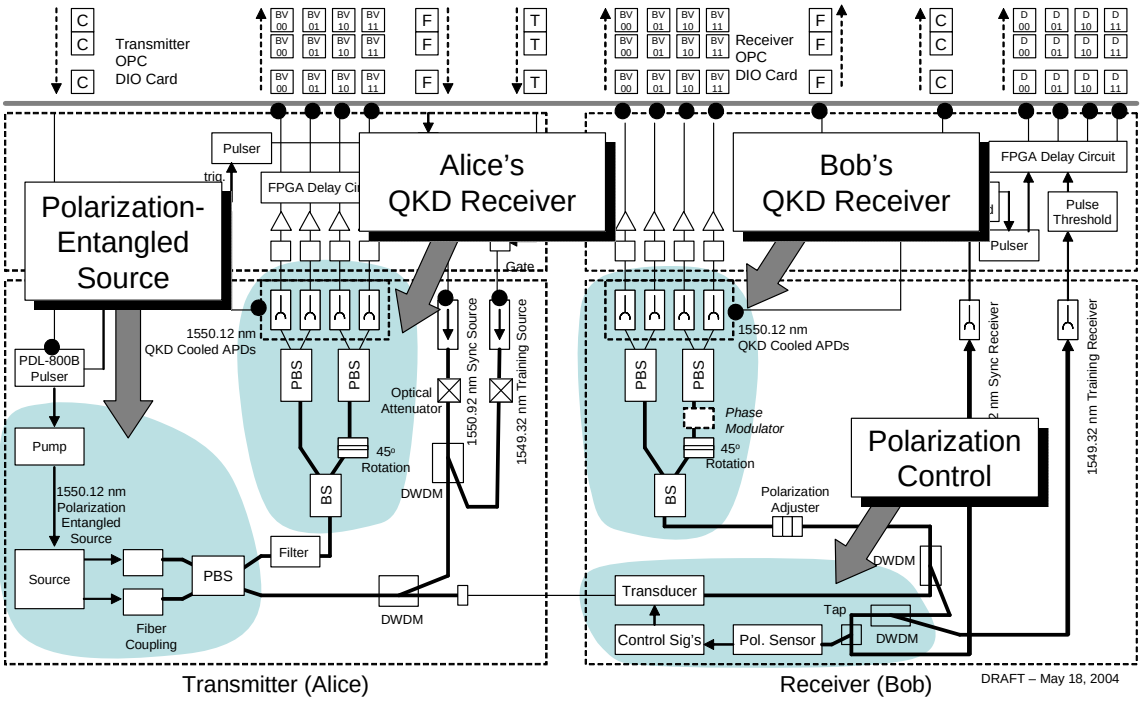


Figure 11-6. Functional Decomposition of the Mark 1 Entangled QKD Link.

All detectors are cooled InGaAs APDs, gated at the expected arrival times of QKD photons. This gating interval is driven by a clock within Alice, which also pulses the source pump. At every pump time, then, Alice gates her detectors. If one (and only one) detector clicks, then Alice assumes she has received one photon from a pair. She then records which detector clicked, giving her (basis, value) pair, and transmits a sync signal (bright pulse) to Bob.

Meanwhile the other entangled photon from this pair is delayed, via a long fiber loop within Alice, so that the sync signal can get to Bob before this entangled photon. When Bob receives this sync signal, he gates his detectors, and collects whatever detection events occur during this gating interval. If any detection event occurred, Bob can then read out a (basis, value) pair. In later stages of the BB84 protocol, Alice and Bob may perform sifting, error detection and correction, and privacy amplification on these recorded (basis, value) pairs.

Note that in our present Mark 1 design, Alice emits a Sync pulse at every pulse interval, and hence Bob gates his detectors for every pulse interval. We may someday change this so that Alice only emits a Sync pulse when she has detected a valid single-photon event; this would reduce Bob's dark count and might lead to better overall system performance.

11.2.2 Polarization Control and Framing for the Mark 1 Entangled Link

Since Alice and Bob employ polarization modulation, and telecommunications fiber acts as a "polarization scrambler," the Mark 1 Entangled Link requires polarization control at Bob to ensure that received pulses are restored to their original polarization before being demodulated and detected. We have chosen to tackle this problem via active polarization control within Bob.

There are two distinct mechanisms needed for polarization control, one for each of the two receiving bases. Figure 11-7 shows those parts of the system most closely tied to the polarization control for receiving pulses in the 0 / 90^{o} basis.

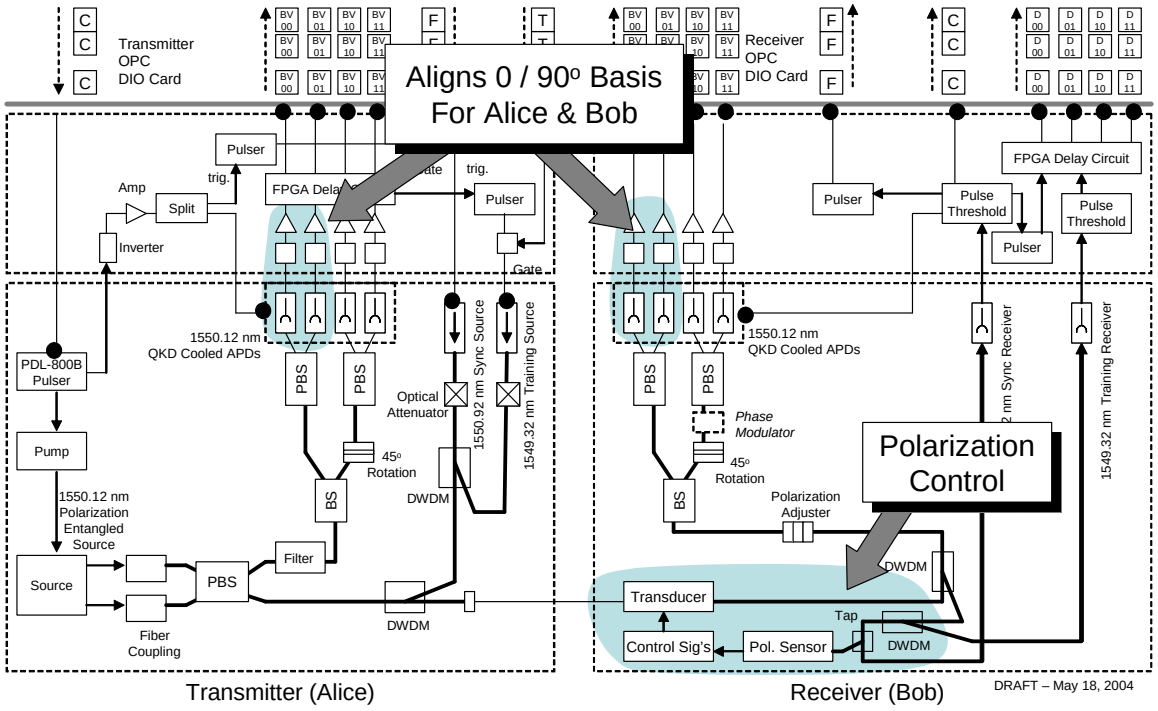


Figure 11-7. Polarization Control for Receiving the 0 / 90 Degree Basis.

Here is how it works. A commercial polarization control is employed within Bob. It takes its input light from the 1550.92 Sync pulses sent by Alice in a known polarization, and restores these sync pulses to the original polarization by splitting off a portion of the light from these sync pulses and using its own sensors and actuators to align polarization for this frequency.

Two aspects of this approach require comment. First, we believe that controlling polarization at the Sync frequency (1550.92 nm) will also adequately control the polarization at the nearby frequency used for the "single photon" QKD pulses (1550.12 nm). We are currently performing experiments to determine whether this is actually true. Second, commercial polarization controllers are generally designed to accept relatively bright, continuous wave light as the incoming optical signal. By contrast, we are driving it with a series of relatively dim pulses of low duty cycle (the Sync pulses). Thus some experimentation or modification to the commercial equipment may be required in order to achieve adequate performance.

In a nutshell, then, the 0 / 90^o basis will be polarization-controlled by a standard controller driven by incoming Sync pulses of a known polarization. This brings us to the second half of the problem, namely of inducing the proper phase shift within the received 45 / 135^o basis once the 0 / 90^o basis has been properly established.

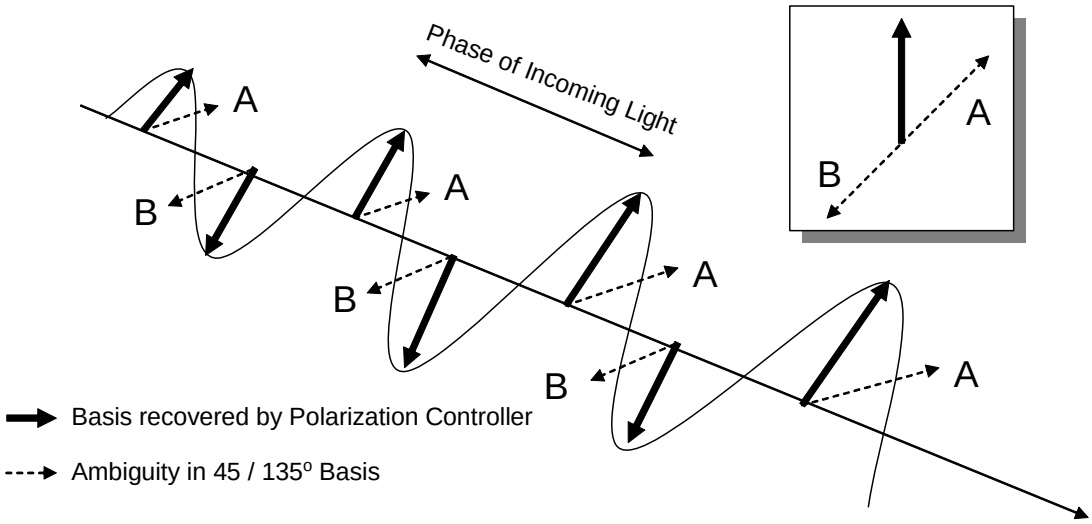


Figure 11-8. Remaining Ambiguity of 45 / 135^o Basis after the 0 / 90^o Basis has been Recovered.

Figure 11-8 diagrams this problem in highly schematic form. Here we can see that even after the Polarization Control has recovered one basis correctly, there are two possible settings for its non-othogpnal 45 / 135^o basis: one correct (A) and the other its inverse (B). Hence the mechanism discussed so far will deliver correct patterns of detector clicks at the leftmost pair of QKD detectors in Bob, for the 0 / 90^o basis. However the patterns of detector clicks will sometimes be received correctly in the rightmost pair, for the 45 / 135^o basis, but they will often be inverted because this basis has been incorrectly recovered.

As can be seen, the phase of the incoming light is related to which basis (A or B) is recovered. Flipping this argument around, we can say that one can recover the correct 45 / 135^o basis by *adjusting* the phase of incoming light pulses. **Figure 10-2** illustrates our proposed mechanisms for locking in on the correct 45 / 135^o basis once the 0 / 90^o basis has been recovered. Here we have interposed a Phase Modulator along the path towards the two detectors for the 45 / 135^o basis. By adjusting the light's phase as it passes through this modulator, we can continuously adjust the light so that it recovers the correct pattern of 0 and 1 detects rather than the inverted pattern.

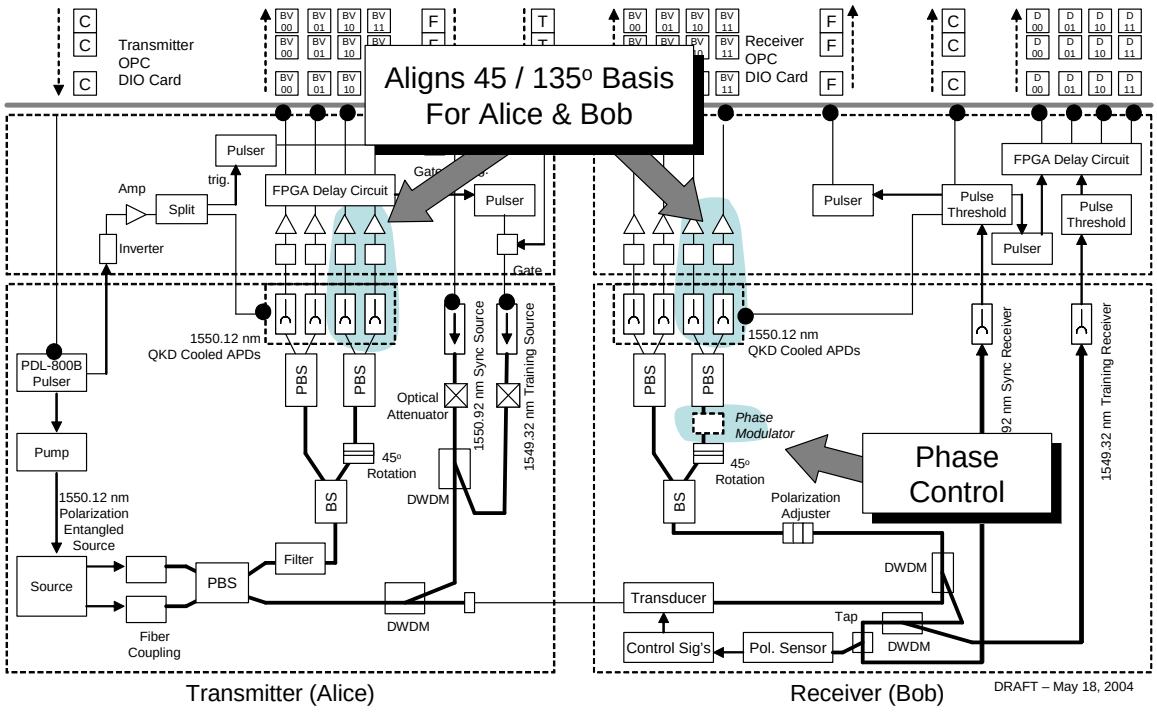


Figure 11-9. Polarization Control for Receiving the 45 / 135 Degree Basis.

However, this begs the question of how we know how to control the phase modulator, e.g., how much voltage should we apply so that we are receiving the "correct" pattern of 0 and 1 detects in this basis. In short, we have thus far discussed the actuator for this control loop, but not the sensors or control algorithms.

The answer to this question is *training frames*. In much the same way that known patterns dim pulses are employed in the Mark 2 Weak Coherent Link, representing 0s and 1s in the various bases, the Mark 1 Entangled System employs training frames to recover correct polarization for the 45 / 135° basis. By using this novel approach, we hope to leverage the control algorithms that we have already developed but employ them in an entirely new way.

Figure 11-10 highlights those parts of the Mark 1 Entangled Link that are closely related to training frames. This mechanism is discussed at length in Section 11, but we briefly sketch it here. First, all training frames consist of random patterns of 0s and 1s in random bases – just as in ordinary (cryptographic) operation of the link. However, Alice designates certain of her transmitted frames as training frames; for such frames, each dim "single photon" pulse is accompanied by a bright Training pulse (1549.32 nm) that encodes which of Alice's detectors clicked, if any, for that pulse. Thus in training frames, each "single photon" pulse is accompanied by a much brighter pulse that clearly announces the (basis, value) for that pulse.

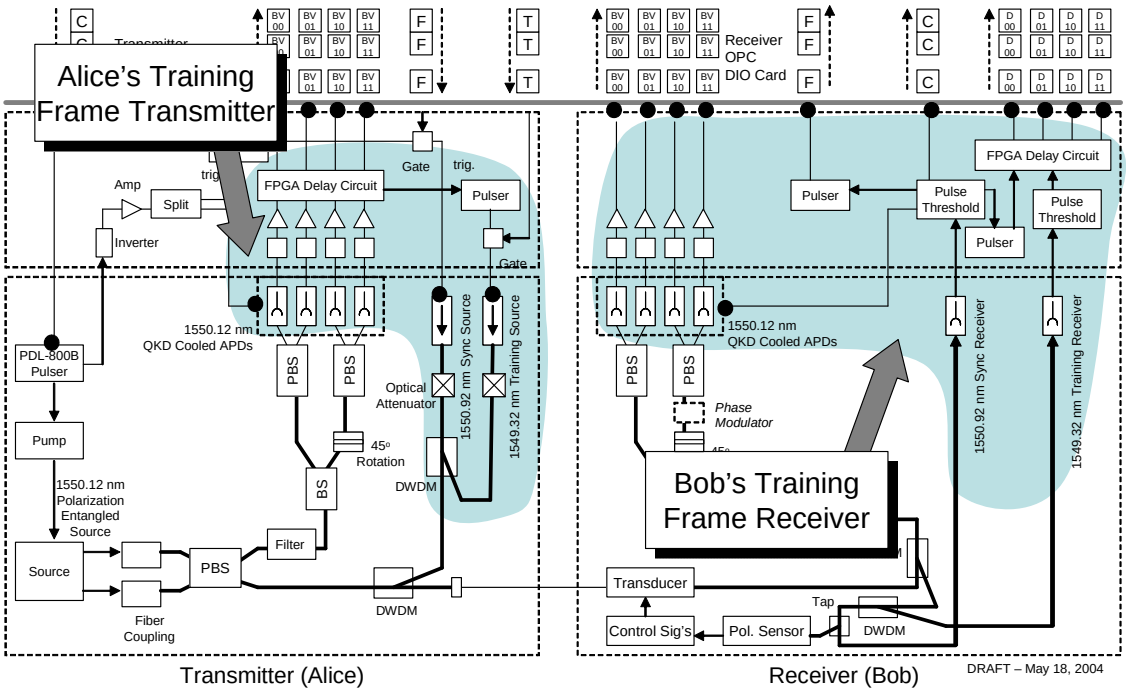


Figure 11-10. Training Frame Transmission and Reception in the Mark 1 Entangled Link.

When Bob receives such training frames, then, he has perfect information about Alice's detection of the (basis, value) for each dim pulse in the frame. If polarization control is working properly, Alice and Bob should have good correlation between their 'value' detections in basis 0 (the 0 / 90^o basis).

More importantly, however Bob can use the training information to control his phase modulator and thus properly recover the 45 / 135^o basis. This can be done by inspecting Bob's own detection statistics for basis 1, and comparing the directly detected results against Alice's corresponding values as encoded in the Training pulses. An algorithm can then minimize the difference between these statistics by adjusting Bob's phase modulator appropriately.

IMPLEMENTATION NOTE: In early operation of the Mark 1 Entangled Link, we will perform this phase modulation manually; it is likely to stay aligned for perhaps a few minutes at a time. We will then implement automatic control of phase modulation sometime after the full system is operational.

IMPLEMENTATION NOTE: We believe it may also be possible to directly control the 45 / 135^o basis through the standard polarization controller (Transducer) itself, in addition to using it for controlling the 0 / 90^o basis. This will require BBN software driving the inputs to the Transducer, based on information derived from received training frames. We will explore this possibility as we build out the system.

11.3 Key Design Decisions for the Mark 1 Entangled Link

This section discusses key design decisions for the Mark 1 Entangled Link. This link has proved challenging to design, in several ways. The major decisions have involved:

- Polarization vs. Phase Modulation

- Operating Wavelength within Alice

11.3.1 Polarization vs. Phase Modulation

Because we desire a continuously operational QKD system, we prefer implementations that run through fiber optic strands. In particular, we want to use existing dark fiber between the BBN, BU, and Harvard campuses for metro-Cambridge networks.

Telecommunications fiber acts as a "polarization scrambler," and thus it is difficult to successfully convey polarization-modulated information (such as QKD pulses) through such a channel. Accordingly, we initially planned to build a phase-modulated system based on Franson interferometers[9]. It appears that such a system has been experimentally demonstrated by the Geneva group.

However, as we began to explore phase-modulated systems in greater detail, it appeared that they are quite difficult to stabilize.

11.3.2 Operating Wavelength within Alice

Another key question was that of the operating wavelength to employ within Alice (i.e. near the BBO crystal). There were two major possibilities: (a) operate at 810 nm within Alice, or (b) operate at 1550 nm within Alice. Bob's detectors were fixed at 1550 nm so as to allow operation through telecommunications fiber.

There is one enormous advantage to an asymmetric setup that runs Alice's detectors at 810 nm, which is that silicon detectors may be employed. These detectors do not need gating; they can operate in "staring" mode and click whenever a photon strikes. They also have relatively high Quantum Efficiency (QE), which will be essential for good throughput. However, they have quite a strong disadvantage as well, namely, the entanglement fidelity with a mixed (810 / 1550) system is considerably lower than that with a symmetric (1550 / 1550) system. The following tables present analysis performed by Boston University in the course of determining final system design.

[9] J. D. Franson, "Bell Inequality for Position and Time," Physical Review Letters, v. 62, n. 19, (8 May 1989).

810 / 1550 nm (Pump at 532 nm)	
Advantages	**Disadvantages**
▪ Higher QKD rate due to higher quantum efficiency and lower noise in Alice's silicon APDs. ▪ 810 nm fiber components are available from OzOptics, etc. ▪ BU has demonstrated 80% visibility with PPLN already. ▪ Can use Si APDs for Alice so 50% fewer Epitaxx InGaAs APDs required	▪ Less symmetric system design ▪ 80% Visibility is a best case result and only intermittently achievable at present. No known demonstration of better visibility. ▪ Estimate 90% Visibility needed for reasonable QBER. ▪ Tedious to adjust & stabilize & align

1550 / 1550 nm (Pump at 775 nm)	
Advantages	**Disadvantages**
▪ More symmetric entangled source design – higher visibilities likely (higher quality of entangled states due to better spectral symmetry). ▪ More symmetric (uniform) entangled QKD link design – more standard telecom components. ▪ "Near co-linear" propagations significantly simplifies photon fiber coupling. ▪ Scalable to "Entangled Source in the Middle" architecture. ▪ Easier transition to longer-term applications such as secret sharing, teleportation, entanglement swapping.	▪ Limitations of InGaAs APD performance. Noise, Q.E., gate pulse shape. ▪ QKD rate suffers due to low detector efficiency at both Alice and Bob ▪ Pulsed pump required, in order to drive gating signals for Alice's InGaAs detectors.

12 Random Numbers – Generation and Testing

This section outlines the critical role of cryptographic-quality random numbers in the DARPA Quantum Network, identifies the sub-components in which random numbers are used, and describes the generation and testing of these random numbers.

The need for truly random numbers is clearly exemplified with the BB84 protocol. If an eavesdropper, Eve, were able to predict one of {Alice's random bits sent, Alice's encoding bases, or Bob's measuring bases} then she would always be able to find the shared secret. Here is why. In the first case, the random bit string sent by Alice is the bit string that will be publicly distilled to the secret shared secret. In the second case, if Eve knew the basis Alice encoded her random bits by and was eavesdropping on the quantum communication channel, then Eve could decode and determine all the random bits Alice sent. Finally, if Eve knew the basis that Bob measured and was eavesdropping on the quantum communication channel, then Eve could determine exactly what Bob measured and distill the shared secret.

Thus the unconditional security of both information theoretic and quantum cryptography rests on the proper use of truly random numbers. As a direct consequence, a solid architecture for producing, testing, and using unpredictable random numbers is paramount to the security of the DARPA Quantum Network.

12.1 Categories of Random Number Generators and Their Appropriate Tests

In theory, truly random numbers are everywhere in nature. In practice, however, creating truly random numbers is as much a philosophical problem[10] as an engineering problem. One can create a device to produce a sequence of random numbers based on a theoretical source of randomness, but the actual device will differ from its theoretical ideal, e.g., by hardware imperfections and possibly malfunctions. How do we know the numbers coming from a device is truly random? The answer is that we will never know. Thus we must create weaker – more attainable -- definitions of randomness. To do so we describe random numbers by their generators.

We will now describe three broad categories of random number sources, along with appropriate tests, each of which may have its distinct use in the DARPA Quantum Network:

- Nondeterministic Random Number Generators (NRNG)
- Cryptographically Secure Pseudorandom Number Generators (CSPRNG)
- Weak Pseudorandom Number Generators (WPRNG)

12.1.1 Nondeterministic Random Number Generators (NRNG)

Definition. A nondeterministic random number generator is based on measurements of unpredictable physical processes. In theory a NRNG produces truly random number because it is based on a theoretical reduction to a truly random source such as thermal noise or measuring a photon in superposition. However real-world concerns such as environmental effects, hardware imperfections or hardware

[10] Testing for true randomness is undecidable. See: J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

malfunctions pose implementation problems. Also imperfections may cause bias that must be de-biased or whitened.

NRNGs are absolutely required in some areas of the DARPA Quantum Network in order to ensure unconditional security. However, current NRNGs are very slow compared to pseudorandom number generators. For the fastest transmissions rates possible, we wish to use NRNGs only when necessary. For instance, NRNG's are imperative in the optical process of the DARPA Quantum Network such as when Alice and Bob select random bases to send and measure randomly encoded qubits.

Tests. We plan to employ the NIST FIPS 140-2 tests[11] for our NRNG. See Section 12.2.

Current NRNG Selection. The DARPA Quantum Network currently employs an RBG1210 random bit generator chip in a BBN SafeKeyper as its NRNG. The RBG1210 chip is based on thermal noise.

12.1.2 Cryptographically Secure Pseudorandom Number Generators (CSPRNG)

Definition. A cryptographically secure pseudorandom number generator is a deterministic algorithm[12] that, given a numeric seed, outputs a sequence that a classical computer cannot differentiate from a truly random sequence in a reasonable (polynomial) amount of time.

CSPRNGs are based on a specific type of mathematical one-way functions, namely trapdoor functions. Trapdoor function based CSPRNG, such as public key cryptosystems, are used most commonly because they can be mathematically proven to be as strong as a hard math problem such as the discrete log problem and the factoring conjecture. However once a general-purpose quantum computer exists, these two hard math problems can be solved efficiently using Shor's Algorithm[13].

A CSPRNG based on a one-way function, such as a hash function based on a symmetric key cryptosystem, is conjectured to be as hard as finding the key to the cipher. Similarly, symmetric key cryptosystems are prone to attacks by quantum computers using Grover's Algorithm[14]. Using a quantum computer to find the key in any 128 bit key symmetric key cryptosystem would typically require $2^{\frac{128}{2}-1} \cong 2^{63}$ queries.

Tests. Testing for randomness in CSPRNGs is well developed. We shall comply to the NIST standard for cryptographically secure pseudorandom number generators[15]. As per the NIST standard, level 4 CSPRNGs must be tested on startup and on demand. In the following sections we will outline the four NIST-mandated tests: the monobit test, the poker test, the runs test and the long runs. The tests are run on

[11] FIPS 140-2: Security Requirements for Cryptographic Modules. National Institute for Standards and Technology. Available at: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.
[12] "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." – John von Neumann, 1963, *Various Techniques Used in Connection With Random Digits*.
[13] P. W. Shor. *Algorithms for quantum computation: Discrete logarithms and factoring*, 35nd Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, pp. 124-134, 1994.
[14] Gilles Bressard. *Searching a Quantum Phone Book*, Science, vol. 275. pp. 627-628. 1997.
[15] FIPS 140-2: Security Requirements for Cryptographic Modules. National Institute for Standards and Technology. Available at: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

the same sequence of 20,000 bits. If any test fails then the generator must be reseeded and retested. Even though the output of a CSPRNG is determined by its seed, we will test our CSPRNG periodically to comply with the NIST standard.

Current CSPRNG Selection. The DARPA Quantum Network currently employs Yarrow using SHA-1 and the AES as its CSPRNG.

12.1.3 Weak Pseudorandom Number Generators (WPRNG)

Definition. A weak pseudorandom number generator is a fast, but predictable, deterministic random number generator. On input of a truly random seed, a WPRNG will provide adequately random numbers for solving some statistical problems. This class of random number generators is extremely fast. WPRNG's are used in where randomness is needed but the randomness does not affect security, such as when selecting random blocks of data during error correction.

Tests. In general, the outputs of WPRNGs need not be tested since they are deterministic given the algorithm and the seed. (We assume that failures of the algorithm execution, e.g., via processor fault or memory bit error, are sufficiently rare so that the actual execution may be considered a perfect implementation of the algorithm.) If the seed is appropriate then the WPRNG will be appropriate for non-cryptographic purposes.

Current WPRNG Selection. The DARPA Quantum Network currently employs the Mersenne Twister as its WPRNG.

12.2 Testing For Randomness

All nondeterministic processes working with random numbers or numbers dependent on a random variable based on NRNG and CSPRNG should be tested. Obviously, we must continually test the NRNG itself. For level 4 FIPS compliance we must test the CSPRNG on startup and on demand too.

The implemented CSPRNG will be tested at startup, on demand, and periodically with both the NIST tests and physical tests. Any NIST test failure will flag a `WARNING` resulting in a reset and repeated test. Three failed tests in a row will result in an `ERROR`. If a physical test fails then an `ERROR` will result.

The NRNG will run the NIST tests and the Universal Statistic Test randomness tests on startup and on demand on all data in its buffer. Any test that fails will result in a `WARNING` and the test will be repeated. Three `WARNING`'s in a row will result in an `ERROR`. The passed data set will be set in a cache to be used by the randomness consumers.

We must also test random numbers that have been processed through non-ideal devices and/or channels. Alice's photonics system, the optical channel and Bob's photonics system may create bias. We cannot continually test the individual devices since measuring the quantum states will alter their value. The only place we can test is after Bob's measurement. In a perfect system, Bob's measurements would have a measured significance level directly related Alice's measured significance levels. In imperfect systems, we still expect Bob's measurements will follow the underlying probability distributions that define these tests---a χ^2 distribution. We can compare Bob's measured significance level to Alice's measured

significance levels to gain information on the quality of the OPC to determine an introduction of bias and flag a possible `WARNING`.

12.2.1 The NIST FIPS 140-2 Tests for Secure Randomness

NIST FIPS PUB 140-2 specifies four tests: the monobit test, the poker test, the runs test and the long runs. The tests are run on the same sequence of 20,000 bits. If any test fails then the generator must be reseeded and retested.

12.2.1.1 The Monobit Test

The monobit test is used to check that the ratio of 1's to 0's is about 1:1. The NIST standard specifically asks to check that the number of 1's is between 9,725 and 10,275 in a 20000-bit block. If the 1's are with the bound then pass, otherwise fail.

12.2.1.2 The Poker Test

The Poker Test tests whether sequences of length *m* appear approximately the same number of times within a sample space of length *n*. For the NIST test, divide the same 20,000 bits into 4-bit blocks, $b_1b_2b_3b_4$, where b_i are bits. Denote $f(b_1b_2b_3b_4)$ as the number of blocks containing the value $b_1b_2b_3b_4$. The following bound must hold,

$$2.16 < \frac{16}{5000}\left(\sum_{i=0}^{15} f^2(b_1b_2b_3b_4)\right) - 5000 < 46.17$$

The NIST test fails if the equation is outside the bounds, and passes otherwise.

12.2.1.3 The Runs and Long Runs Test

A *run* is defined as a maximal sequence of consecutive bits of either all 1's or all 0's. A *gap* is a run of all 0's and a *block* is a run of all 1's. This test determines if the number of runs is as expected for a random sequence. As per the NIST test: runs are calculated in the 20000-bit sample block. The runs must have a specified spacing between them--an interval. The NIST intervals are defined as:

Run Length	Required Interval
1	2315 to 2685
2	1114 to 1386
3	527 to 723
4	240 to 384
5	103 to 209
6 to 25	103 to 309
26 or more (Long Runs)	No long runs may be present

To clarify, any run greater or equal to 6 must be separated by 103 to 209 other runs. Also there must be no runs greater than 26. The NIST tests pass if all bounds are met, otherwise it fails.

12.2.2 Maurer's Universal Statistical Test

To be supplied (TBS).

12.3 Randomness Requirements in the DARPA Quantum Network

In this section we list where randomness is required throughout the DARPA Quantum Network, identify the rate in which the randomness is required, and identify the minimal required type of random number generator required to create the randomness.

Figure 12-1 depicts the approximate rate at which various components of the DARPA Quantum Network will consume random numbers, measured in bits / second. This is an initial estimate and will be refined as we gain more experience with the actual system operation. As can be seen, however, the highest-rate consumers are the optical processes: Alice's source requires 2 random bits per laser pulse (or approximately 10,000,000 bits/second), and Bob's requires half that amount (about 5,000,000 bits/second).

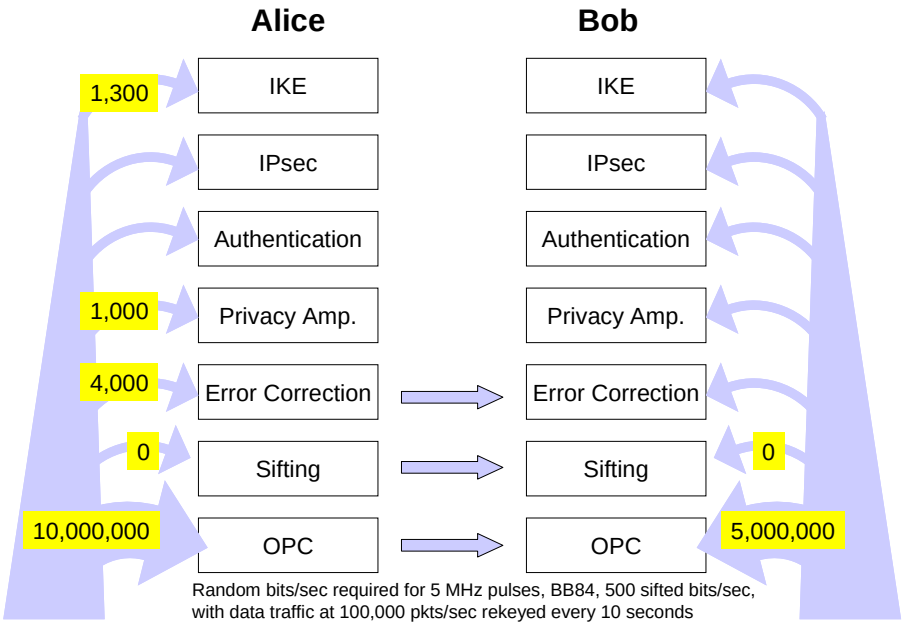


Figure 12-1. Approximate Rates of Random Bit Consumption in Weak Coherent Link.

The following sections discuss these rates in detail, i.e., present an initial analysis – on a per-subsystem level – of the rate at which random numbers will be consumed in the DARPA Quantum Network with the first Weak Coherent Link.

12.3.1 Randomness Required in the Photonics Subsystem

To implement the BB84 the *Source Suite* (Alice) requires 2 bits from a NRNG per photon being sent. That is a bit rate of 2 times the pulse rate on Alice's side. Similarly, the *Detector Suite* requires 1 bit from

a NRNG per photon expected to be received, a bit rate of 1 times the pulse rate on Bob's side. Anything less random than the NRNG would compromise the unconditional security and would open the system to exploitable flaws. If we expect the single photon pulse rate to be 5MHz, then the *Source Suite* requires 10Mb/s of NRNG bits and the *Detector Suite* requires 5Mb/s of NRNG bits.

12.3.2 Randomness Required in the BBN QKD Protocols

First, *Error Correction* requires 63n bits per QFrame from a WPRNG. More generally we will need the following number of WPRNG bits.

(number of blocks)*(the number of sifted measurments)
= (number of blocks)*(p*(size of raw QFrame))

Where p is the probability of Bob measuring a sent qubit. Each QFrame will require a 32-bit reseeding from an NRNG.

Second, *Privacy Amplification* must use nondeterministic random numbers. *Privacy Amplification* is purely based on a random hash function to surprise an eavesdropper so she cannot formulate an attack prior to the application of the hash function. If Eve knew of the hash function we were going to use then she could perform a successful attack to drastically weaken the key strength. The number of random bits required is equivalent to the size of the bit string being amplified.

12.3.3 Randomness Required in the IPsec Protocol Suite

Currently, we need random numbers for Internet Key Exchange (IKE) protocol, the AES initial vector (IV), and for padding. Since these random numbers are used for computationally secure cryptographic purposes, a CSPRNG fits this purpose best.

IKE requirements may be separated into two phases. Phase 1 requires randomness for a 64 bit cookie, two Diffie-Hellman half keys (1024 bits each) and an RSA signature (500 bits). The rate equations for all three are unknown. The cookie need only be from a WPRNG, while the others should be from a CSPRNG. Phase 2 requires a 256 bit nonce, a 1024 bit Diffie-Hellman half key and a 16 bit qblock id. All random bits in IKE must come from a CSPRNG.

The rate equations for the IV and padding are unknown. They have been approximated at 1Mb/s for the IV and padding.

In the future, IPsec will implement the one time pad. Pads will arrive directly from the quantum key distribution system. One-time pads do not use IV's. “Padding” strings are not necessary since the one time pad is unconditionally secure.

12.4 Current Implementation of Randomness in the DARPA Quantum Network

As discussed above, the most demanding consumers of random numbers in our current system are Alice’s and Bob’s photonics subsystems at roughly 10 and 5 million random bits / second respectively. These sequences need to be of the highest grade, i.e., nondeterministic random numbers.

Unfortunately, it is difficult to produce random numbers at these rates. In fact, the fastest NRNG that we know runs at approximately 1 million bits/second, and thus is a factor of 10 too slow for our system. Furthermore we do not have such a system at hand.

Our current randomness strategy for the DARPA Quantum Network is as follows. We attach a BBN SafeKeyper NRNG to both Alice and Bob, which produces roughly 9,600 random bits / second. We then "expand" this sequence to the required rate (e.g. to 10 million bits / second) by using the nondeterministic sequences as a continuous series of seeds to a CSPRNG. This produces about 1,000 bits of cryptographic randomness per 1 bit of nondeterministic randomness, i.e, expands our physically random sequence by this factor.

When faster NRNGs are available for our use, we will remove the relatively slow SafeKeyper from our system and plug in the faster NRNG in its place.

12.5 Terms Used in this Chapter

Item	Description
Nondeterministic Device/Process	A device or process in which its output is not determined by it's input. A computer is deterministic process. A truly random number generator must be nondeterministic.
NRNG	A Nondeterministic Random Number Generator is a nondeterministic process or devices that outputs a sequence of random bits. We use the RGB1210 chip basing randomness on thermal noise.
PRNG	A Pseudorandom Number Generator is a deterministic process or device that on input of a small random sequence of bits, will create a large amount of output that looks random to all deterministic processes.
WPRNG	A Weak Pseudorandom Number Generator is a PRNG that produces adequate randomness for statistical purposes. However, one can predict the future random bits based on the prior output of a WPRNG. We use the Mersenne Twister WPRNG.
CSPRNG	A Cryptographically Secure Pseudorandom Number Generator is a PRNG where it is computationally *hard* to predict the future output based on the prior output. We use the Yarrow design with SHA-1 and AES.
Randomness Test	One or many algorithms to check for problematic sequences of random bits.
χ^2 distribution	χ^2 (chi squared) distribution is a robust predicting mechanism to identify the outcome of many repeated random events.
Goodness-of-fit	Goodness-of-fit is a mathematical measurement to quantify how closely an observed sequences of events match with an abstract probability distribution, such as a χ^2 distribution.

13 The BBN QKD Protocols

This section presents a technical introduction to the QKD protocols and algorithms implemented in the DARPA Quantum Network. It describes each of the protocols and algorithms in high-level form and shows how these individual sub-components are assembled to form an entire QKD protocol stack. This section also describes the interface between the BBN QKD Protocols and the IPsec protocol suite.

13.1 "Eve" and "Mallory" Terminology in this Document

In the usual terminology of classic cryptography, "Eve" generally implies a third party that may perform eavesdropping but not fabricate or alter any communications between Alice and Bob. The nickname "Mallory" is often used for a malicious third party that can modify, delete, and fabricate messages between Alice and Bob.

In this document, we follow the slightly different terminology widespread in the quantum cryptography community, in which Eve can take on all the attributes of Mallory, and in which there is thus no explicit mention of Mallory. Thus in this document, Eve is presumed capable of modifying, deleting, and fabricating messages between Alice and Bob – on both the photonic channel and on the "public" channels for the QKD protocols – in addition to pure eavesdropping.

In particular, we must protect against cases in which Eve performs "man in the middle" attacks, i.e., where Eve interposes herself between Alice and Bob so that all of Alice's conversations are in fact carried out with Eve instead of Bob, with a similar interposition for Bob's conversations. These attacks are parried by use of continuous, mutual authentication of Alice and Bob, as described below.

13.2 Terminology for QKD Protocols and Algorithms

Figure 13-1 presents our terminology for the sub-functions involved in QKD protocols and algorithms, along with the terminology for finished product available at the end of each of these functions. This terminology has not yet been standardized in the literature, but our terminology does reflect widespread usage.

Note that we do not refer to any of the intermediate or finished products as "keys" since (a) these QKD bits can be used for other purposes than cryptographic keys, and (b) in the Quantum Network, these bits are not themselves used as keys but rather form only one part of the raw material that the IPsec mechanisms turn into keys.

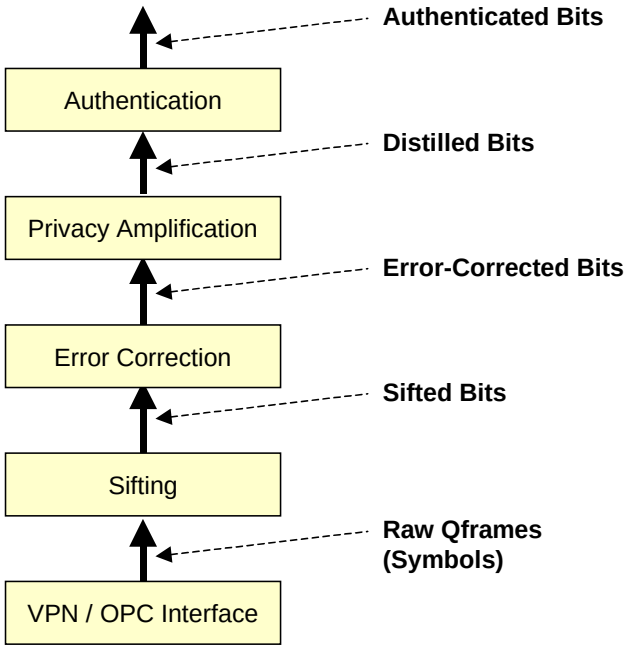


Figure 13-1. Terminology for the Sub-Layers of QKD Protocol Suite.

We describe each of these functions in some detail in the subsequent sections, with the exception of the lower-layer VPN / OPC interface (Section 10.9) and its product, Raw Qframes (Section 10.6). These elements properly belong to the physical layer of the overall QKD protocol stack and so are described in other portions of this document.

13.3 A Simple Introduction to QKD Protocols and Algorithms

Figure 13-2 gives some flavor of the most basic operations performed by the QKD protocols and algorithms. At the bottom of the figure, we see the raw qubits as prepared and transmitted by Alice (labeled with Tx for transmit) and the related photo-detections observed at Bob (labeled Rx for receive). Reading up the figure, we see how these bits are first sifted to eliminate various types of transmission and reception error, and then the resulting sifted bits are error-corrected to remove bits that were "damaged in transit."

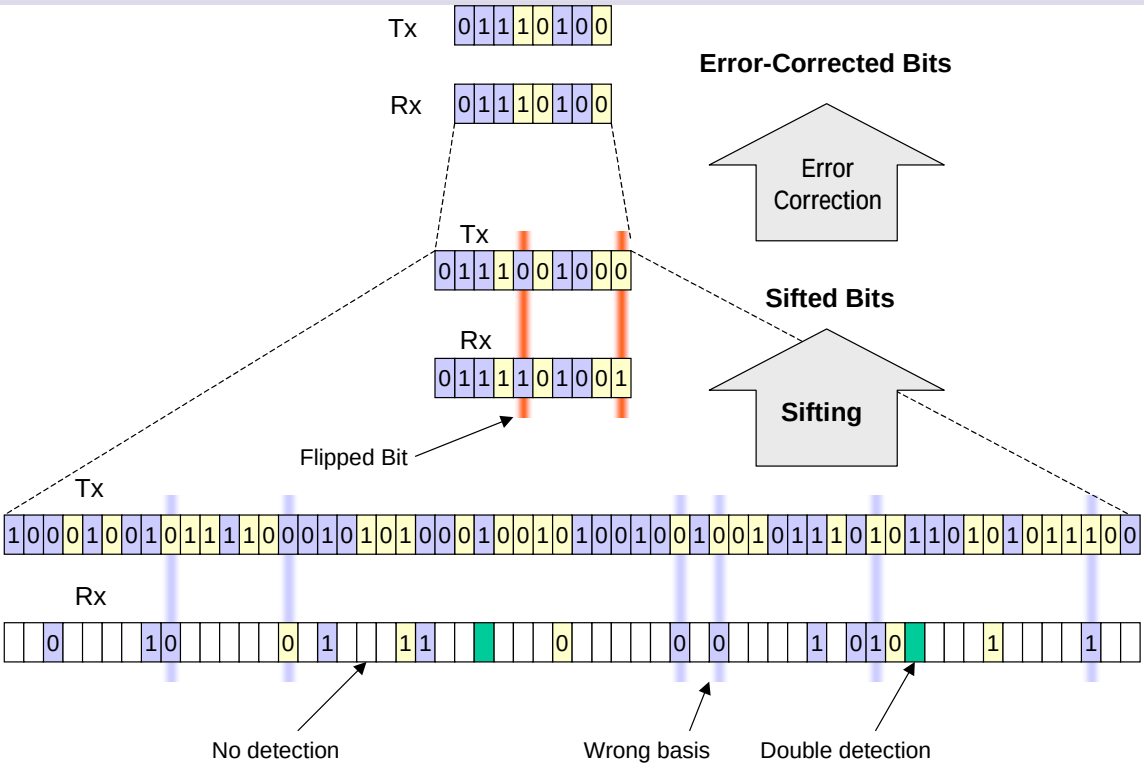


Figure 13-2. The Functionality Provided by Sifting and Error Correction.

As shown in the figure, sifting is the process by which Alice and Bob reconcile the contents of their Raw Qframes to remove the following types of errors:

- No detection symbols, where Alice believed that she prepared and transmitted a photon but neither of Bob's detectors fired. Such events are very common and have several distinct causes. Some of the most common are: (a) Alice's source did not actually emit a photon; (b) that photon was lost in transmission; (c) Eve captured the photon and did not replace it; (d) Bob's detector didn't fire when the photon hit it.

- Wrong basis symbols, where Bob did receive a detection but in fact Alice and Bob did not use the same basis for this bit, i.e., the basis that Alice randomly selected was different from that which Bob randomly selected.

- Double detection symbols, in which both of Bob's detectors fired. This makes it impossible for Bob to determine whether the symbol should be treated as a 0 or a 1, and so the symbol must be discarded.

The sifting process involves both communication (QKD protocol interactions) between Alice and Bob, and internal algorithms within Alice and Bob. Both are implemented via software within the Quantum Protocol Daemon. The sifting process is described in some detail below, but in its essence Bob tells Alice the basis he used when receiving valid symbols, and Alice replies with a list of those symbols for which she used the same basis. Then Alice and Bob discard all other symbols and keep only the resulting "sifted" bits.

Once Alice and Bob have agreed on sifted bits, they must perform error correction to find and then eliminate those bits that have been damaged in transmission, i.e., the flipped bits that were sent as a 1 but received as a 0 or vice versa. There are many ways to perform error correction but each has two important consequences:

1. Error correction is always probabilistic – unless all bits are revealed during the process! -- and thus there is some possibility that Alice and Bob will believe that they share an identical set of bits, but in fact they do not.

2. Error correction requires communications between Alice and Bob, and inevitably – assuming that Eve can obtain plain text versions of all such public communications – the process of error correction reveals information about the sifted bits to Eve.

Thus the end result of error correction is two-fold. First, Alice and Bob will with high probability contain, in their local memories, identical copies of a set of error-corrected bits. Second, Eve will have some knowledge of these bits' values, which must then be reduced by the subsequent process of privacy amplification.

Privacy amplification allows Alice and Bob to reduce Eve's knowledge of their error-corrected bit values to an arbitrarily low fraction of the total number of such bits. Unfortunately it also reduces the number of shared bits in the process, and it does not reduce Eve's knowledge to zero! At its core, it is implemented by randomized algorithms within Alice and Bob, and public communication between the two as to the results of these algorithms. Our implementation of privacy amplification is discussed in detail in a section below.

Finally, authentication allows Alice and Bob to have reasonable assurance that they are in fact communicating with each other. In most modern cryptographic systems, this function is performed by some kind of one-way functions, e.g., digital signatures implemented by public key techniques. In the classic literature of quantum cryptography, authentication relies on shared secret keys, e.g., in universal hash functions. Such shared secrets must be pre-staged (in a very familiar chicken-and-egg scenario) and then refreshed from time to time via newer bits obtained through the quantum channel. The quantum cryptographers' approach seems to have several different weaknesses when viewed in a system-wide sense, but of course is not vulnerable to attacks based on quantum computing or mathematical breakthroughs such as fast factoring. We expect that we will explore this aspect of quantum cryptography at some length during the course of the Quantum Network project.

Item	Description
Authentication	Authentication is the process by which Alice and Bob guard against "man in the middle attacks," i.e., by which Alice determines that she is communicating with Bob (and not Eve) and vice versa. Authentication must be performed continuously, rather than once at startup, in order to protect against cases in which Eve inserts herself into a conversation between Alice and Bob after the initial authentication has been performed.
Authenticated Bits	Authenticated bits are those secret bits shared by Alice and Bob that have been authenticated, i.e., for which Alice has high assurance that the bits are actually shared with Bob (rather than an imposter such as Eve) and

Privacy Amplification	Privacy amplification is the process whereby Alice and Bob reduce Eve's knowledge of their shared bits to an acceptable level. This technique is also often called distillation.
Distilled Bits	Distilled bits are those secret, shared bits produced by Privacy Amplification. To a very high probability, Eve should have no knowledge of these bits' values.
Error Correction	Error correction is the process whereby Alice and Bob determine all the "flipped bits" among their shared, sifted bits, and correct them so that Alice and Bob share the same sequence of error-corrected bits. Flipped bits are ones that Alice transmitted as a 0 but Bob received as a 1, or vice versa. Note that these bit errors can be caused by noise or by eavesdropping.
Error-Corrected Bits	Error-corrected bits are those secret, shared bits produced by Error Correction. To a very high probability, they should be identical within Alice and Bob.
Sifting	Sifting is the process whereby Alice and Bob winnow away all the obvious "failed qubits" from a single Raw Qframe or a series of Raw Qframes. As described in the introduction to this section, these failures include those qubits where Alice's laser never transmitted, where Bob's detectors didn't work, where photons were lost in transmission, and so forth. They also include those symbols where Alice chose one basis for transmission but Bob chose the other basis for receiving.
Sifted Bits	Sifted bits are those secret, shared bits produced by Sifting. Note that these bits may be corrupted by noise (i.e. different at Alice and Bob) and that Eve may have some knowledge of the possible values for these bits.
Raw Qframes	A Raw Qframe is a fixed-size block of symbols transmitted from an OPC to its VPN. At the transmit side, it contains an indication of the bases and values that were prepared for each outbound single-photon qubit. At the receive side, it contains the basis and detector-hit values for each inbound qubit. See Section 10.6 for a description of how Raw Qframes reflect the underlying physical processes on the QKD link. The "VPN / OPC" Interface Control Document defines the exact format of Raw Qframes.
VPN / OPC Interface	The interface between the QKD protocol engine (housed in the VPN computer) and the Optical Process Control entity housed in a separate computer. See Sections 9.5.1 and 10.9.
Flipped Bit	Flipped bits are those bits that Alice transmitted as a 0 but Bob received as a 1, or vice versa. Note that these bit errors can be caused by noise or by eavesdropping.
No detection	A "no detection" symbol indicates that neither of Bob's QKD detectors fired when an incoming single-photon was expected from Alice.
Wrong basis	A "wrong basis" symbol occurs when Alice and Bob determine that they did not randomly agree on the same basis for a given qubit, and hence that Bob did not correctly modulate his receive suite to receive this qubit.
Double detection	A "double detection" symbol indicates that both of Bob's QKD detectors fired when an incoming single-photon was expected from Alice.

13.4 QKD "Shared Secrets" are Neither Perfectly Shared nor Perfectly Secret

The astute reader will have uncovered two important facts in the course of reading the previous section. Despite all claims (hype) to the contrary, the keys established by QKD techniques are in the general case neither perfectly shared, nor perfectly secret!

They are not perfectly shared, since noise in the channel can only be corrected probabilistically. Good choice of error correction codes can give an asymptotically small probability of Alice and Bob having different "shared" bits after error correction, but it cannot totally eliminate the possibility. Thus a reliable system must be able to resynchronize its cryptographic tunnels in cases where Alice and Bob disagree on key material. (Of course, in practical systems this can also happen because bits become corrupted elsewhere in the system, e.g., within Alice's memory while awaiting use.)

They are not perfectly secret, because the knowledge that Eve learned from the error detection and correction protocol cannot be completely eliminated, although it can be reduced to an arbitrarily small ratio compared to the final number of shared secret bits. Making this ratio very small does have a high expense, though, as it drastically reduces the available number of shared secret bits.

Thus a better formulation of the benefits of quantum cryptography might be: QKD allows pair-wise agreement upon secret, randomized bits with an arbitrarily high probability of those bits having identical values, and an arbitrarily low probability of any third party knowing those values. However, the higher the amount of assurance desired, the lower the rate (bits/sec) at which these bits can be agreed upon.

13.5 Prior Work in QKD Protocols and Algorithms

There has been a great deal of prior work in QKD protocols and algorithms, starting when the field was first invented by Charles Bennet and Gilles Brassard in 1984[16], and independently, at about the same time by Artur Ekert[17]. Their ideas, in turn, were based on an earlier proposal by Stephen Wiesner from the 1970s. We recommend the review article "Quantum Cryptography," by Nicolas Gisin et al.[18], as an excellent introduction to the subject. It is clear and thorough, with a fine list of references.

BBN researchers owe a large debt of gratitude to many individuals and organizations who have worked to shape this field. Among these, our strongest influences to date have come from the work at Los Alamos and IBM Almaden. Indeed, our first weak-coherent link has been patterned upon our understanding of the mid-1990s Los Alamos fiber-based prototype (though we believe that our design has a number of significant differences from preceding designs).

Figure 13-3 is work in progress. In it, we attempt to categorize all known implementations of the QKD protocols and algorithms. There are significant variants of some of the sub-functions. This is especially notable at the Quantum Encoding layer, but is also true for error correction and privacy amplification.

---

[16] Bennet, C. H. and G. Brassard, 1984, "Quantum cryptography: public key distribution and coin tossing," Int. conf. Computers, Systems & Signal Processing, Bangalore, India, December 10-12, 175-179. This paper introduces the field and also defines the BB84 protocol that we implement in the Quantum Network.

[17] Ekert, A.K. "Quantum cryptography based on Bell's theorem," Phys. Rev. Lett. 67, 661-663. This article was published in 1991 based on earlier work.

[18] Gisin, N., Ribordy, G., Tittel, W., and H. Zbinden, "Quantum cryptography," quant-ph/0101098, January 19, 2001.

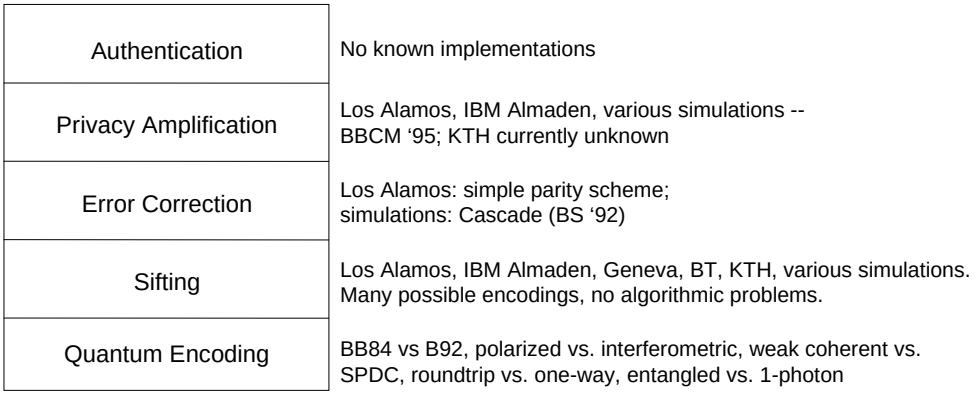


Figure 13-3. QKD Protocols and Algorithms, with their Known Implementations.

13.6 QKD Implementation in the Year 1 Quantum Network

This section describes the protocols and algorithms that we intend to implement by the end of Contract Year 1 for the DARPA Quantum Network. Note that we expect to design and implement additional protocols in subsequent years, but that we have not yet identified this future work in any detail.

QKD Sub-Function	Specific Techniques	Description
Authentication	Universal Hash Function	Preposition a "small" shared secret key at Alice and Bob, and use this key as input to a Universal Hash function along with the refined bits obtained by QKD protocols. Use the result as a cryptographic checksum to verify Alice or Bob's identify.[19,20]
	Hybrid Public Key / Universal Hash Function	Combine the Universal Hash Function approach with Public Key Cryptography, i.e., digital signatures. While this approach does not satisfy the purist quantum cryptographic community, it may be a good engineering approach for communities that believe that public key techniques are still useful.
Estimation of Eve's Knowledge	BBSSS	$\frac{4e}{\sqrt{2}}$, std. deviation $\sqrt{(4+2\sqrt{2})e}$. Taken from Bennet et al[21]; see Section 13.9 for a discussion.
	Slutsky	$(b-e)\left[1+\log_2\left(1-\frac{1}{2}\left(\frac{1-3e'}{1-e'}\right)^2\right)\right]$, std. dev. $\sqrt{b}$ Taken from Slutsky et al[22]; see Section 13.9 for a discussion.
Privacy Amplification	Universal Hash Function based on Rényi Entropy	Use a hash function to reduce the size of a batch of error-corrected, shared secret bits by an amount sufficient to reduce Eve's possible knowledge of the resultant bits' contents to a sufficiently small amount (e.g. far less than 1 bit's worth).[23]
Error Correction	Parity Checks	A conventional parity-checking scheme as widely employed in telecommunications systems.[24]
	Cascade	Select random subsets of the sifted bits, compute and exchange parity bits on a subset to detect errors, and then use a divide-and-conquer scheme to correct any detected errors.[25]

[19] This form of authentication scheme was introduced in C. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," International Conference on Computers, Systems, and Signal Processing, Bangalore, India, 1984. It was based on Wegman-Carter hashes as referenced below.

[20] Universal hash functions were introduced in M. Wegman and L. Carter, "New Hash Functions and their Use in Authentication and Set Equality," J. Comp. Sys. Sci., 22, 265-279 (1981).

[21] Bennett, C. H., Bessette, F., Brassard, G., Salvail, L. and Smolin, J., "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, 1992, pp. 3 – 28

[22] Slutsky, B., Rao, R., Sun, P. C., Tancevski, L. and Fainman, S., "Defense frontier analysis of quantum cryptographic systems," *Applied Optics*, vol. 37, no. 14, 1998, pp. 2869 – 2878.

[23] The usual reference for this technique is C. Bennett, G. Brassard, C. Crépeau, and U. Maurer, "Generalized Privacy Amplification," 1995.

[24] Described in Richard Hughes et al, "Quantum cryptography over underground optical fibers," in N Koblitz, editor, *Advances in Cryptology -- CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 329-342, 18-22 August 1996. Springer-Verlag.

25 Described in G. Brassard and L. Salvail, "Secret key reconciliation by public discussion," Lect. Notes in Computer Science 765, 410. (1994)

Sifting	Run-Length Encoding	Encode the sifting messages, as sent between Bob and Alice, efficiently so that runs of identical values (and in particular of “no detection” values) are compressed to take very little space.
Quantum Encoding	BB84, Interferometric, Weak Coherent Source	Use a “classic” phase encoding, via matched Mach-Zehnder interferometers, with a weak coherent source, suitable for operation through telecom fibers.

Note that the exact encodings employed in the BBN QKD protocol messages is a protocol design issue, and hence will not be discussed in this document. Please refer to the “QKD Protocols and Algorithms” document for this level of detail.

13.7 Sifting

Sifting is the process whereby Alice and Bob winnow away all the obvious “failed qubits” from a single Raw Qframe or a series of Raw Qframes. As described in the introduction to this section, these failures include those qubits where Alice’s laser didn’t emit a photon, where Bob’s detectors didn’t work, where photons were lost in transmission, and so forth. They also include those symbols where Alice chose one basis for transmission but Bob chose the other basis for receiving.

Figure 13-4 shows this process in schematic outline, emphasizing the relationship between the lower-layer frames of transmit or receive symbols (i.e. Raw Qframes) and the sifting process.

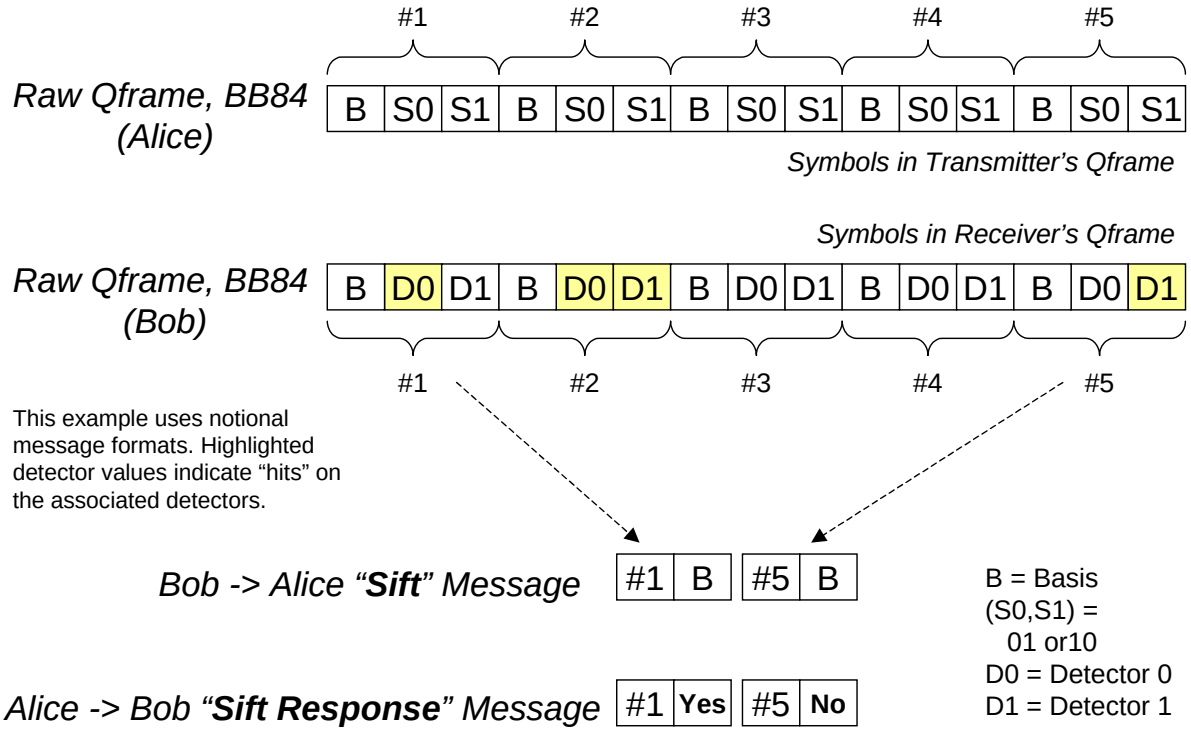


Figure 13-4. Sifting – A Schematic View.

We expect that Bob's raw Qframes will be relatively sparse for the Mark 2 Weak-Coherent link, i.e., Bob's detectors will seldom fire at all. Thus most of Bob's received symbols will likely be blanks. This is because we will attenuate Alice's 1550 nm QKD source laser until it very rarely emits more than a single photon. An unavoidable side-effect for such types of lasers is that this level of attenuation means that they rarely produce even a single photon when run at such low levels. Thus in the majority of cases, Alice will prepare a state but then never actually emit a photon. On the other hand, Bob's detectors may be noisy enough so that we receive "hits" fairly often even when no photon was transmitted, due to dark noise in the detectors. So it is possible that Bob's Raw Qframes will not be sparse after all. We will not know the answers to such questions until we have had a chance to perform systematic experiments with our optical equipment.

However that may be, Bob's software looks through one or more of its received Raw Qframes and eliminate all erroneous symbols (i.e. blanks and double-firings). It then formulates a QKD protocol "sift" message that indicates the symbols it actually received, by frame number and symbol offset within the frame number, along with the basis it used for each of these received symbols. It then transmits this message to Alice.

When Alice receives this message, she looks through the symbol offsets listed in the message and finds the corresponding symbols in her transmitted Raw Qframes. For each such symbol, she checks Bob's basis (as given in the sift message) with her own basis, and formulates a "Yes" or "No" reply depending on whether the bases match or don't match, respectively. Then she sends back this list of Yes and No replies in a "sift respond" message, and the protocol interaction is finished.

At the end of this round of protocol interaction – i.e. after a sift and sift response transaction – Alice and Bob discard all the useless symbols from their internal storage, leaving only those symbols that Bob received and for which Bob's basis matches Alice's.

In general, we expect the sifting phase to dramatically prune down the number of symbols held in Alice and Bob. For instance, let us assume that 1% of the photons that Alice tries to transmit are actually received at Bob and that the system noise rate is 0. On average, Alice and Bob will happen to agree on a basis 50% of the time in BB84. Thus only 50% x 1% of Alice's photons give rise to a sifted bit, i.e., 1 photon in 200. A Raw Qframe of 1,000 bits therefore would boil down to about 5 sifted bits. Note once again that the 1% number is purely notional; we expect to start getting real values for our weak-coherent link in Summer 2002.

The run-length encoding used by BBN's sifting protocol is not highly sophisticated, but is designed to efficiently represent bit strings from fully random (hence not compressible) to extremely sparse. Each entry in the run-length encoding is a single byte or a number of bytes. There are encodings for a small number of zero bits followed by a one bit, a large number of zero bits, or a verbatim encoding for dense regions (so the encoding will never be much worse than simply listing a bitstring, but it will generally be much better).

13.8 Error Correction

Error correction is the process whereby Alice and Bob determine all the "flipped bits" among their shared, sifted bits, and correct them so that Alice and Bob share the same sequence of error-corrected bits. Flipped bits are those bits that Alice transmitted as a 0 but Bob received as a 1, or vice versa. Note that these bit errors can be caused by noise or by eavesdropping.

At least two different error-correction techniques have been discussed in the quantum cryptographic literature, as listed in the following table.

Technique	Description
Cascade	Selects random subsets of the sifted bits, computes and exchanges parity bits on a subset to detect errors, and then uses a divide-and-conquer scheme to correct any detected errors. Described in G. Brassard and L. Salvail, "Secret key reconciliation by public discussion," *Lect. Notes in Computer Science* 765, 410. (1994)
Simple Parity	A conventional parity-checking scheme as widely employed in telecommunications systems. Described in Richard Hughes et al, "Quantum cryptography over underground optical fibers," in N Koblitz, editor, *Advances in Cryptology -- CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 329-342, 18-22 August 1996. Springer-Verlag.

Figure 13-5 depicts in simple form how parity-checking techniques XOR together different subsets of the shared, sifted bits to form a single 0 or 1 which gives the parity of this subset. Multiple such subsets, generally arranged so as to be overlapping in various ways, give rise to multiple parity bits. These bits are then transmitted from Bob to Alice (or vice versa) via a public channel.

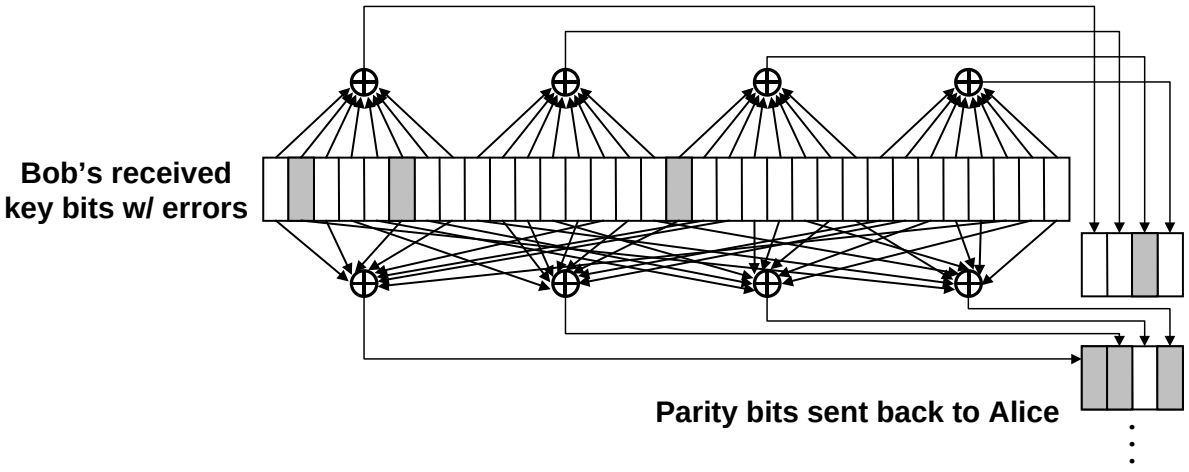


Figure 13-5. Error Correction – A Schematic View.

The recipient then performs an identical series of XOR operations on its own local copy of the sifted bits to obtains its own version of the parity bits. These bits are then matched against the received copy. If they are identical, it is assumed that the two copies of sifted bits are identical. Otherwise the two copies are

clearly different, and some form of error correction must then be brought into play. We should finish this process with:

1. The same strings on both sides, with very high probability[26]
2. An accurate count of the number of bits disclosed
3. A count of the number of bits in error

Our first approach for error correction is a BBN-designed variant of the Cascade protocol and algorithms. The protocol is adaptive, in that it will not disclose too many bits if the number of errors is low, but it will accurately detect and correct a large number of errors (up to some limit) even if that number is well above the historical average.

BBN's version works by defining a number of subsets (currently 32) of the sifted Qframe—one of which is the entire frame—and forming the parities of each subset. The process begins with the transmitter (whoever sent the last protocol message, but in the sifting protocol described above, that's the transmitter). This allows the messages to be batched, and reduces the number of round-trip delays. In the first message, the list of subsets and their parities is sent to the other side, which then replies with its version of the parities. The subsets are pseudo-random bit strings, from a Linear-Feedback Shift Register (LFSR) and are identified by a 32-bit seed for the LFSR.

Then we begin a process of reconciling the two parity sets. When all parities agree, we can quit. The algorithm is similar to the BINARY procedure in Cascade, except that the endpoints take turns choosing the subranges and piggyback the reply on the next subrange parity. During the course of reconciliation, we keep parities for all the subsets and for the known subranges of those subsets. If it's our turn to speak, and not all parities agree, we choose the smallest subset on which we know there's a disagreement, break it in half, and send the other end the parity of the first half. He will compute his parity, decide if the error was in that half or the other, and break that piece in half, and so forth. When we reduce the discrepancy to a single bit, one side or the other changes it (it doesn't matter who, since we only want the two strings to agree).

Once the bit has been flipped, both sides go through their records of all the subsets and their subranges, and flip the recorded parity of those that contained that bit. This will clear up some discrepancies but may introduce other new ones, and so the process continues.

Since these parity fields are revealed in the interchange of "error correction" messages between Alice and Bob, these bits must be taken as completely visible to Eve. Therefore, the QKD protocol stack makes an internal annotation of how many bits have been revealed (lost) due to parity fields, and will require a compensating level of privacy amplification in order to reduce Eve's knowledge back to acceptable levels.

[26] This area still awaits detailed quantitative design as to optimal overall system behavior, but as a rough estimate the bit-error rate (BER) after error correction should be no more than 10^-6 and probably more like 10^-9.

13.9 Estimates of Eve's Knowledge

Privacy amplification depends on having an estimate of the eavesdropping-free entropy of the Qframe—the amount of information in the frame beyond what Eve might know. The estimate is made after the Qframe has passed through sifting and error correction, and any randomness or bias testing. The inputs to entropy estimation are:

- *b*, the number of received bits (sifted)
- *e*, the number of errors in the sifted bits
- *n*, the total number of bits transmitted
- *d*, the number of parity bits disclosed during error correction
- *r*, a non-randomness measure from randomness tests

The components of the entropy estimate are:

- An estimate of the information Eve possesses due to non-transparent (error-inducing) observations.
- An estimate of the information Eve might possess due to transparent eavesdropping—observations that have no effect on the error rate, e.g. beamsplitting attacks, interceptions of multi-photon pulses, and the like.
- The amount of information disclosed publicly during error detection and correction.
- An estimate of the information Eve might possess due to non-randomness in the raw QKD bits (detector bias, for example).

Of these components, only the third—publicly disclosed information—is clear and non-controversial: it is precisely the number of sets of bits whose parities have been disclosed. The fourth—the non-randomness measure—is only a placeholder at the moment, until randomness testing is put into the system. We assume that this testing will produce an measure in the form of a number of bits by which to shorten the string.

Information from transparent eavesdropping is not uniformly treated, and in fact an honest accounting would leave most current QKD systems with no usable entropy. This category includes all eavesdropping that doesn't cause errors, which was originally thought to include only beamsplitting attacks on multi-photon pulses. It is now clear that there are more general attacks of the same ilk. For instance, Brassard et al. have pointed out[27] that all weak coherent systems are particularly vulnerable to attacks based on a POVM. The amount of information leakage can be proportional to the number of transmitted bits times the multi-photon probability, rather than the number of bits received by Bob. A similar attack works for B92. With BB84 using a SPDC-based source, the amount of information Eve may obtain is only proportional to the number of *received* bits times the multi-photon probability. In order to accommodate both types of sources, we count the entropy loss as a constant m_1 times the total number of transmitted bits plus a different constant m_2 times the number of received bits. These constants are parameterizable based on the multi-photon probability.

[27] Brassard, G., Mor, T. and Sanders, B. C., "Quantum cryptography via parametric downconversion," *quant-ph/9906074*.

The most difficult issue is the amount of information Eve obtains due to non-transparent eavesdropping. Following Slutsky et al.[28], we will call these defense functions. There have been several defense functions published for quantum cryptographic systems. Two of the best known are from Bennett, et al.[29] and Slutsky et al. Neither of these appears to be completely accurate—Bennett's estimate does not take into account all the information Eve can get from indirect attacks that give an error rate less than 25%, and while Slutsky's estimate may be asymptotically correct, it is overly conservative for finite-length blocks.

Because we expect to refine the entropy estimate in future, and because we want to allow comparisons of different systems under like assumptions, we will have a choice of defense function. Initially, there will be two functions available, the one that Bennett, et al. and the defense frontier function for BB84 from Slutsky et al.

Both of these estimates include a margin for certainty based on the standard deviation—in Bennett's estimate, this is 5 standard deviations, including the standard deviation of the multi-photon probability, and in Slutsky's case it is parameterizable in terms of probability of a successful attack, but doesn't include multi-photon probabilities. For consistency, we will separate out the standard deviation of each term and combine them at the end, times a confidence parameter *c* (a parameter *c* of 5 would mean 5 standard deviations, or about 10^{-6} chance of successful eavesdropping).

The defense functions are:

Bennett: $\frac{4e}{\sqrt{2}}$, std. deviation $\sqrt{(4+2\sqrt{2})e}$

Slutsky: $(b-e)\left[1+\log_2\left(1-\frac{1}{2}\left(\frac{1-3e'}{1-e'}\right)^2\right)\right]$, std. deviation $\sqrt{b}$

where $e' = \frac{e}{b} + \frac{c}{\sqrt{b}}$

Putting this all together[30], we combine the different forms of information leakage as follows:

Let *t* be the output of the configured defense function, and *s* be its standard deviation. The entropy estimate will be:

[28] Slutsky, B., Rao, R., Sun, P. C., Tancevski, L. and Fainman, S., "Defense frontier analysis of quantum cryptographic systems," *Applied Optics*, vol. 37, no. 14, 1998, pp. 2869 – 2878.
[29] Bennett, C. H., Bessette, F., Brassard, G., Salvail, L. and Smolin, J., "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, 1992, pp. 3 – 28.
[30] "What a thing it is to see the order which prevails throughout his business! By means of this he can at any time survey the general whole, without needing to perplex himself in the details. What advantages does he derive from the system of book-keeping by double entry! It is among the finest inventions of the human mind . . ." J.W. von Goethe (1749–1832), *Wilhelm Meister's Apprenticeship*. The Harvard Classics Shelf of Fiction. 1917. Fragment of a speech by Werner to Wilhelm, denouncing Wilhelm's poetry; note that Wilhelm demurs.

$$b - r - d - t - m_1 n - m_2 b - c\sqrt{s^2 + m_1 n + m_2 b}$$

Naturally, if this number is less than zero, then zero should be used instead.

13.10 Observations on Renyi Entropy in QKD

This section contains several observations on the role of Rényi entropy in QKD, and how improved QKD throughput might be obtained by selecting an optimal Rényi order on a per-transaction basis. These observations are extracted from a paper by Myers, Wu, and Pearson[31]. To our knowledge, the observations have not yet been rigorously reviewed by independent researchers, and thus their verity is not firmly established.

Originally, QKD privacy amplification was described only in terms of Rényi entropy of order 2. Cachin showed[32] that it can be performed based on the Rényi entropy of any order $R > 1$ in almost exactly the same way, except that the "safety margin" – the extra bits removed to achieve the given security parameters – must be slightly higher for the same parameters. The general formula for the final string length is:

$$F_u(k,n,\epsilon,R) - q - \nu - \log_2(m+1) - \frac{R}{R-1}(t + \log_2 R) - s - \log_2 (R-1) - 2$$

where m is chosen so that $m - \log_2(m+1) = n + t$ and s and t are the security parameters. In our QKD system, we choose these parameters so that $s = t = -\log_2\varepsilon$.

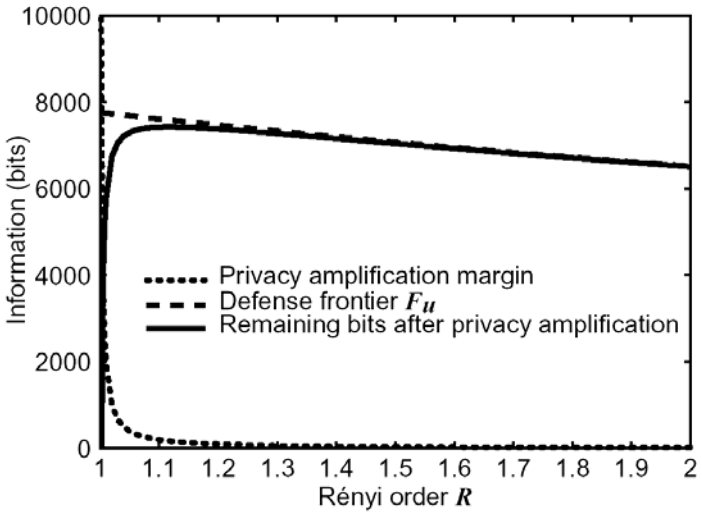


Figure 13-6. Privacy Amplification Calculation for n=10,000, k=500, $\varepsilon = 10^{-6}$, and $s = t = -\log_2\varepsilon$.

Figure 13-6 shows the behavior of $Fu(k,n,\varepsilon,R)$, the privacy amplification margin, and their difference, the number of bits resulting from privacy amplification, as a function of R in a specific scenario. For this graph, Eve's information is assumed to come only from the single-photon pulses, i.e., q and v are zero. As

[31] John M. Myers, Tai Tsun Wu, and David Pearson, "Entropy estimates for individual attacks on the BB84 protocol for quantum key distribution," SPIE, April 2004, Orlando.
[32] C. Cachin, "Smooth entropy and Rényi entropy," Lecture Notes in Computer Science 1233, pp. 193–208, 1997.

can be seen, the defense function *Fu* is monotonically decreasing in *R*, so smaller values of *R* are generally favorable for Alice and Bob. However, the necessary privacy amplication margin increases rapidly when *R* is close to one, and there is an optimum *R* value that yields Alice and Bob the most privacy-amplified bits. (In this figure, the optimum *R* is approximately 1.12.) This optimum *R* can be easily found by Newton's method.

The optimum value of *R* is not fixed, however. It varies with the block size n and the number of error bits *k*. Since $Fu(k,n,\varepsilon,R)$ is a lower bound on Eve's ignorance of the sifted key for any *R*, we can choose the *R* that maximizes the final key size for the observed *n* and *k*. Figure 2 (a) shows how the optimum value of *R* varies with error rate for a fixed block size of 10,000 bits, and figure 2(b) shows how the optimum varies with block size at a fixed error rate of 5%.

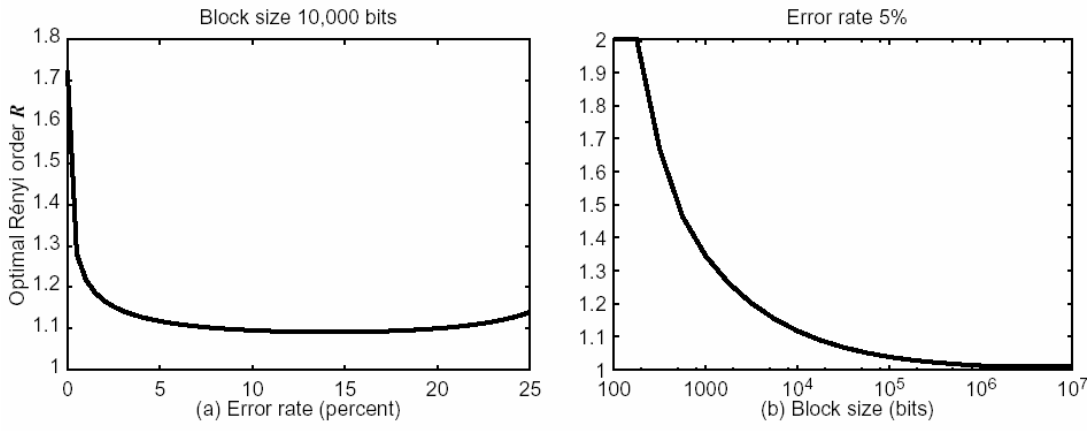


Figure 13-7. How the optimal Rényi order R varies with (a) block size and (b) error rate, $\varepsilon = 10^{-6}$.

Many working QKD systems use the original estimate of Eve's knowledge found in the article describing the first experimental QKD system by Bennett et al.,) even though it is known that this estimate does not represent Eve's optimal attacks. In part, this is because it makes a convenient standard basis for comparison of QKD systems, in part because Slutsky's estimate is less well-known, and perhaps in part because Slutsky's estimate results in significantly lower QKD performance. Figure 3 shows that lower performance is not a necessary by-product of using a more rigorous entropy estimate. It shows the number of bits remaining as a fraction of the block size after privacy amplification, using the estimates of Bennett et al., Slutsky et al., and this article, with varying error rate (a) and block size (b). Again, Eve's information is assumed to come only from the single-photon pulses, i.e., q and v are zero.

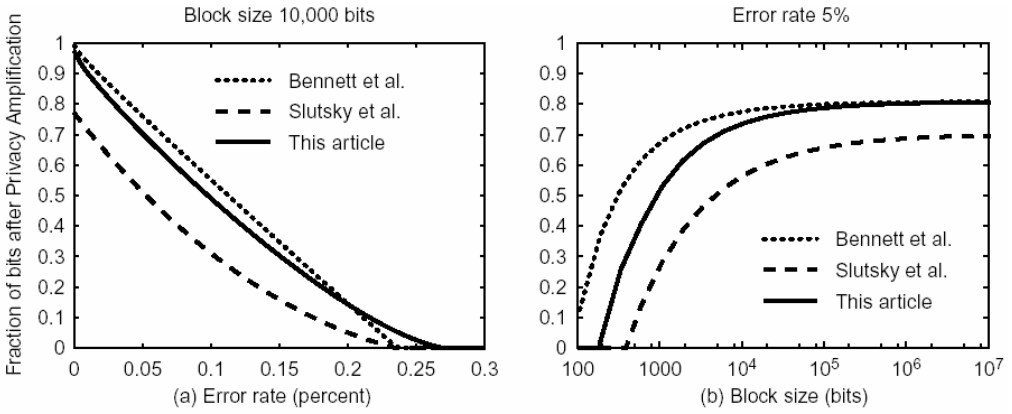


Figure 13-8. Comparison of this estimate with Bennett et al. and Slutsky et al. with varying (a) error rate, and (b) block size, $\varepsilon = 10^{-6}$.

13.11 Privacy Amplification

Privacy amplification is the process whereby Alice and Bob reduce Eve's knowledge of their shared bits to an acceptable level. This technique is also often called distillation or advantage distillation. Please recall that Eve may acquire her illicit knowledge in several ways, such as by eavesdropping below the noise threshold, and/or by observing bits exchanged in the process of error correction. In some cases, Eve may actually know the exact values of specific bits held by Alice and Bob; but more likely, Eve has only some indications that certain families of sequences are more likely than others.

Privacy amplification is a purely classical algorithm that operates on bits in computer memory, and that "smears out" the value of each initial shared bit across the shorter resulting set of bits. Assuming that the "smearing" works properly, one can reduce Eve's knowledge as desired by picking the size of the resulting set of bits. The shorter the resulting set of bits, the less Eve knows. Unfortunately, though, Eve's knowledge can never be reduced to zero by this technique – unless one throws away all the bits! Still, Eve's knowledge can be made arbitrarily small.

Figure 13-9 shows our basic approach to privacy amplification in the DARPA Quantum Network. We employ a universal hash function as our tool for "smearing" bits, giving it as inputs a public random hash and the (shared) secret bits. Its output will be a set of distilled (i.e. privacy-amplified) bits, distributed with equal probability across the resulting space independent of the values of the original shared secret bits.

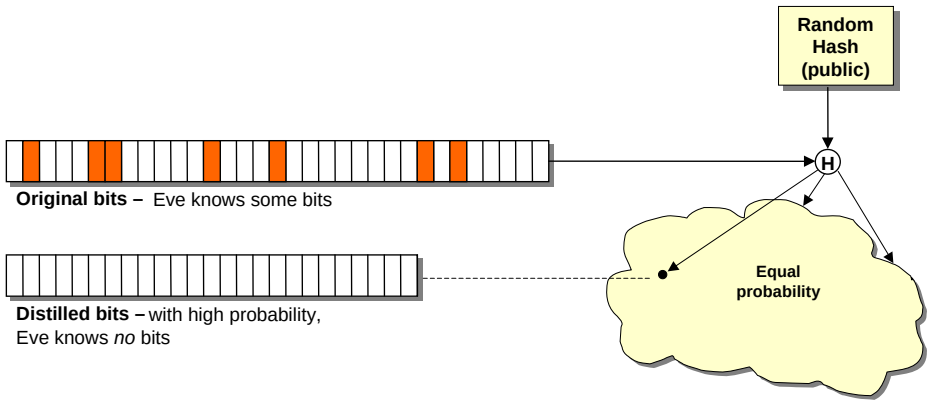


Figure 13-9. Privacy Amplification - A Schematic View.

The privacy amplification protocol has the simple job of specifying a universal hash function to apply to the sifted, error-corrected Qframe, to reduce the number of bits to match the entropy.
The number of bits to reduce is based on an estimate of the number of bits of information an eavesdropper may know about the Qframe, which is at least the number of parity bits disclosed in error correction plus some multiple of the number of bits in error.

The side that initiates privacy amplification chooses a linear hash function over the Galois Field GF[2^n] where n is the number of bits in the Qframe, rounded up to a multiple of 32. He then transmits four things to the other end—the number of bits m of the shortened result, the (sparse) primitive polynomial of the Galois field, a multiplier (n bits long), and an m-bit polynomial to add (i.e. a bit string to exclusive-or) with the product. Each side then performs the corresponding hash and truncates the result to m bits to perform privacy amplification.

13.12 Authentication

Authentication is the process whereby Alice and Bob assure themselves that, with very high probability, they are really exchanging information with each other and not with Eve. There are two basic requirements for authentication in this system. First, it must be *mutual* so that Alice has high assurance that she is indeed talking to Bob, and Bob has similar assurance about Alice. Second, it must be *continuous* so that all parts of the ongoing interactions between Alice and Bob are protected, and not just an initial handshake.

Within the quantum cryptographic community, Eve is generally understood to be limited only by the known laws of physics, and to otherwise possess engineering and mathematical powers far beyond the current state of the art. In particular, it is axiomatic that Eve can:

- Instantly break all mathematical "one-way" functions.
- Detect all dim pulses with zero loss.
- Create dim pulses that are indistinguishable from Alice's except for the limitations of quantum physics (e.g. the no-cloning law).
- Transport photons to Bob with zero loss.
- Etc etc.

It will be seen that, given these basic axioms, Eve can launch highly formidable "man in the middle" attacks against Alice and Bob since she can interpose herself along the photonic channel between Alice and Bob in ways that are very hard to detect, and instantly defeat all ciphers on the "public channel" except one-time pads. Thus authentication plays a major role in the security of a full quantum cryptographic system.

Our approach to authentication follows the original prescription for authentication in quantum cryptography, namely, Universal Hash Functions. This basic approach was very clearly described in the BB84paper on quantum cryptography: "The need for the public (non-quantum) channel in this scheme to be immune to active eavesdropping can be relaxed if the Alice and Bob have agreed beforehand on a small secret key, which they use to create Wegman-Carter authentication tags for their messages over the

public channel. In more detail the Wegman-Carter multiple-message authentication scheme uses a small random key to produce a message-dependent 'tag' (rather like a check sum) for an arbitrary large message, in such a way that an eavesdropper ignorant of the key has only a small probability of being able to generate any other valid message-tag pairs. The tag thus provides evidence that the message is legitimate, and was not generated or altered by someone ignorant of the key. (Key bits are gradually used up in the Wegman-Carter scheme, and cannot be reused without compromising the system's provable security; however, in the present application, these key bits can be replaced by fresh random bits successfully transmitted through the quantum channel.)"

Figure 13-10 presents this approach in graphic form. Note that we employ these technique continuously to ensure that all public channel communications are adequately protected against attacks.

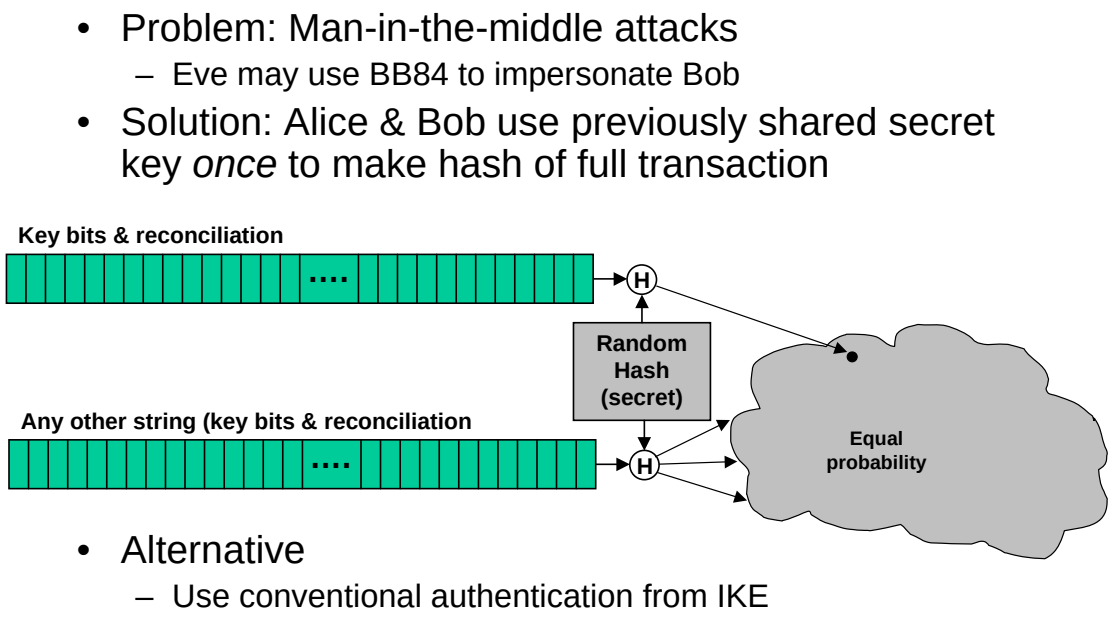


Figure 13-10. Authentication – A Schematic View.

The ratio of hash size to key bit block size determines the probability that Eve will be able to successfully forge the correct hash value. We have not yet determined what probability is suitable for the DARPA Quantum Network, but note that because this test is probabilistic, there is always a non-zero (though small) probability that Eve may launch a successful man-in-the-middle attack.

13.13 Our Unified BBN QKD Protocols

Protocol designers may have noticed, from the following sections, that there appear to be many different back-and-forth transactions between Alice and Bob as they gradually boil down their Raw Qframes to distilled or authenticated bits. Indeed, the QKD protocols must be performed in series rather than in parallel since each step assumes that shared secret bits that have been developed by the previous steps. A series of such "round trip" transactions is undesirable in a networking sense, however, since it takes time and the finished bits will not become available for use until a number of such round trips.

Thus a key goal of the BBN QKD Protocols must be to reduce the number of such round-trips to a minimum. We have made a start at this task in the protocols that we implement for Year 1. To our knowledge, this way of organizing and implementing the QKD protocols is entirely novel. At a high level, our QKD Protocol Suite provides a reliable, in-order, datagram-oriented transaction service between two peers, e.g., Alice and Bob. Each datagram can convey one or more application-level messages. Our design philosophy is spiritually akin to Application Level Framing (ALF)[33].

Figure 13-11 presents an introductory example of our BBN QKD Protocols in action. Here we see a number of messages, in high level schematic form, exchanged in public communications between Alice and Bob. As can be seen, for our weak-coherent link Bob initiates the transactions, and Alice's activities and responses are driven solely by commands sent from Bob.

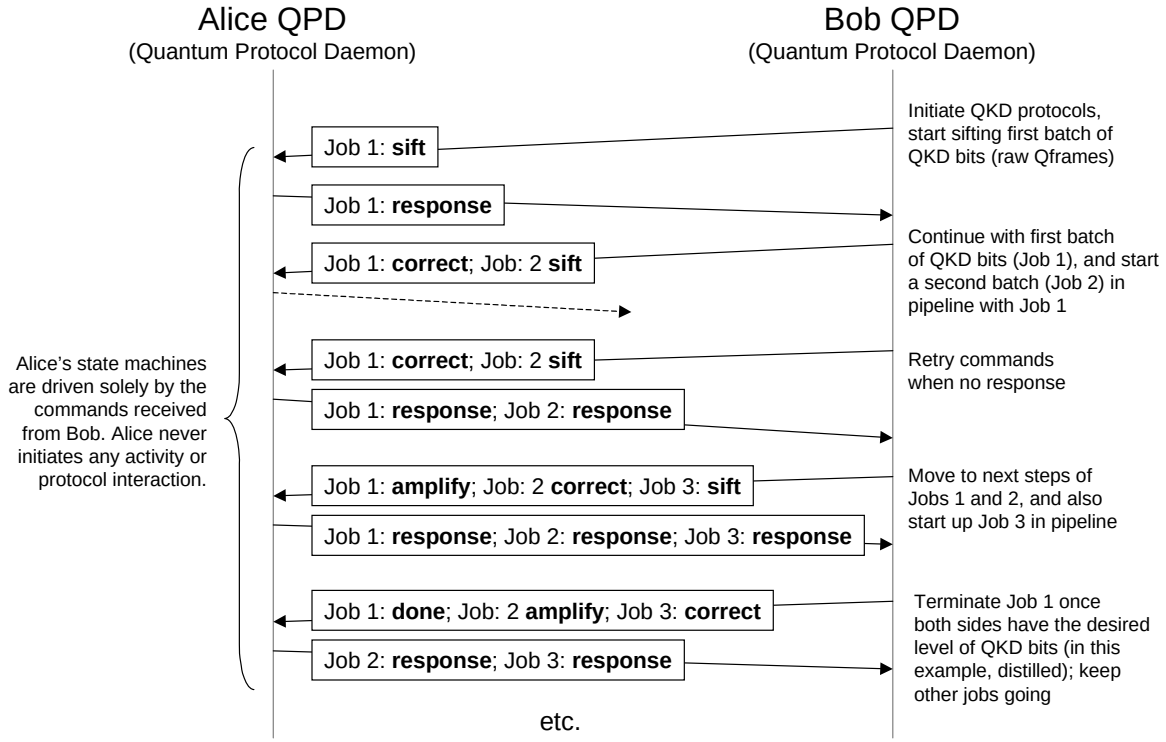


Figure 13-11. Example of our BBN QKD Protocols in Action.

Stepping through this example from the top, we see that Bob initiates one job (Job 1) to start the sifting process on Raw Qframes that have accumulated at Alice and at Bob. Bob can sift either a single Raw Qframe or a series of them in a single message. In response to Bob's message, Alice performs the actions needed for sifting and sends back her response. Meanwhile, both Bob and Alice have accumulated more Raw Qframes, and so Bob starts up a second job (Job 2) in parallel with Job 1. This second job will encompass the processing of the second batch of Raw Qframes. From this time onward, Jobs 1 and 2 are handled in a pipelined fashion so that Bob's error-correction commands for Job 1 are batched with the

[33] D. Clark and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," ACM SIGCOMM, 1990.

sifting commands for Job 2, and so forth. As shown, Bob also starts up a third job and runs all three in parallel until Job 1 is finished. Then Bob shuts down Job 1 but keeps running the later jobs.

This basic process keeps running throughout the entire time that Alice and Bob are exchanging Raw Qframes. Bob creates a new job every time he has a sufficient number of Raw Qframes for starting a new round of processing, and shuts down jobs when the processing of that set of bits is complete.

Figure 13-12 presents the basic outline of the datagram structure for our BBN QKD Protocols. As shown, each datagram begins with a packet header containing all the housekeeping details that allow the protocol to perform reliable, in-order transmission of messages, successfully detect crash and restart of peers, and so forth. The datagram then contains one or more variable-length messages, where each message contains all the information needed in order to describe one command or response for a job. Note that version information is included in these messages because we expect that we will be inventing a number of such protocols during the course of the Quantum Network project, and will likely upgrade our existing protocols as well.

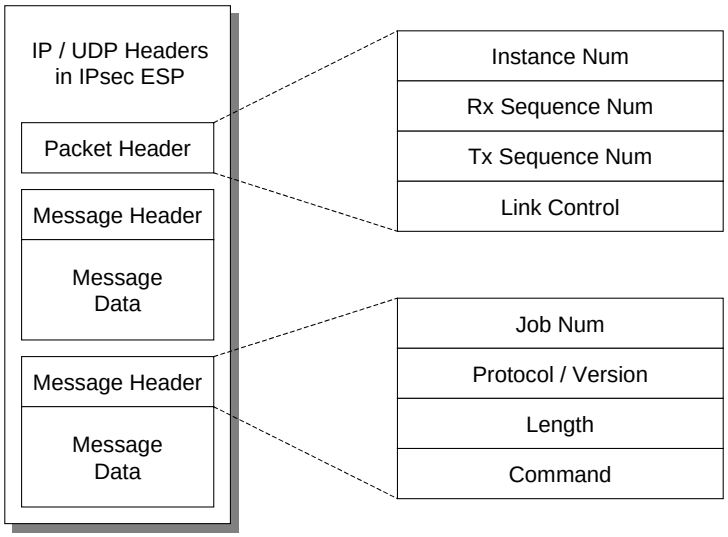


Figure 13-12. High-Level Schematic of the BBN QKD Protocol's Datagram Format.

Note further that we intend the BBN QKD Protocols interactions to be protected by IPsec in normal operation. Thus a secured tunnel will be established between the Quantum Protocol Daemons (QPDs) in Alice and Bob, and all "public" traffic between these two peers will in fact be encrypted, authenticated, and integrity-checked. Of course, it goes without saying that this protection will be turned off during many of our Eve experiments so that Eve can have ready access to the plain text of these "public" messages.

For detailed information on these protocol and algorithm definitions, please consult the "QKD Protocols and Algorithms" document.

13.14 The IKE / QKD Interface

This section describes the software interface between the Internet Key Exchange (IKE) Daemon and the Quantum Key Distribution (QKD) Daemon, both of which software entities run in the same VPN computer. This interface is defined by the "IKE / QKD Interface Control Document," which should be consulted for a further level of details.

This interface provides the following functionality:

1. Establish, maintain, and if necessary re-establish communications between the IKE and QKD Daemons as the system starts up, daemons crash, etc.

2. Reserve identified blocks of shared secret bits (as established by QKD in two different endpoints) so that the IKE Daemon can negotiate over these identified blocks with its peer.

3. Transport identified blocks of shared secret bits from the QKD Daemon to the IKE Daemon as requested by the IKE Daemon. (These bits are then deleted from the QKD Daemon.)

4. Provide miscellaneous set-up and monitoring mechanisms by which the IKE Daemon can determine relevant facts about the QKD Daemon's peers.

Figure 13-13 presents the basic outline of the interactions between the IKE Daemon and the QPD Daemon. Although we expect that both software entities will be resident on the same (VPN) computer, the interface is defined in terms of a reliable "call and response" between the two entities. This message passing may be implemented in any convenient way, e.g., via CORBA, some other form of Remote Procedure Call, explicit messaging via TCP connections, etc.

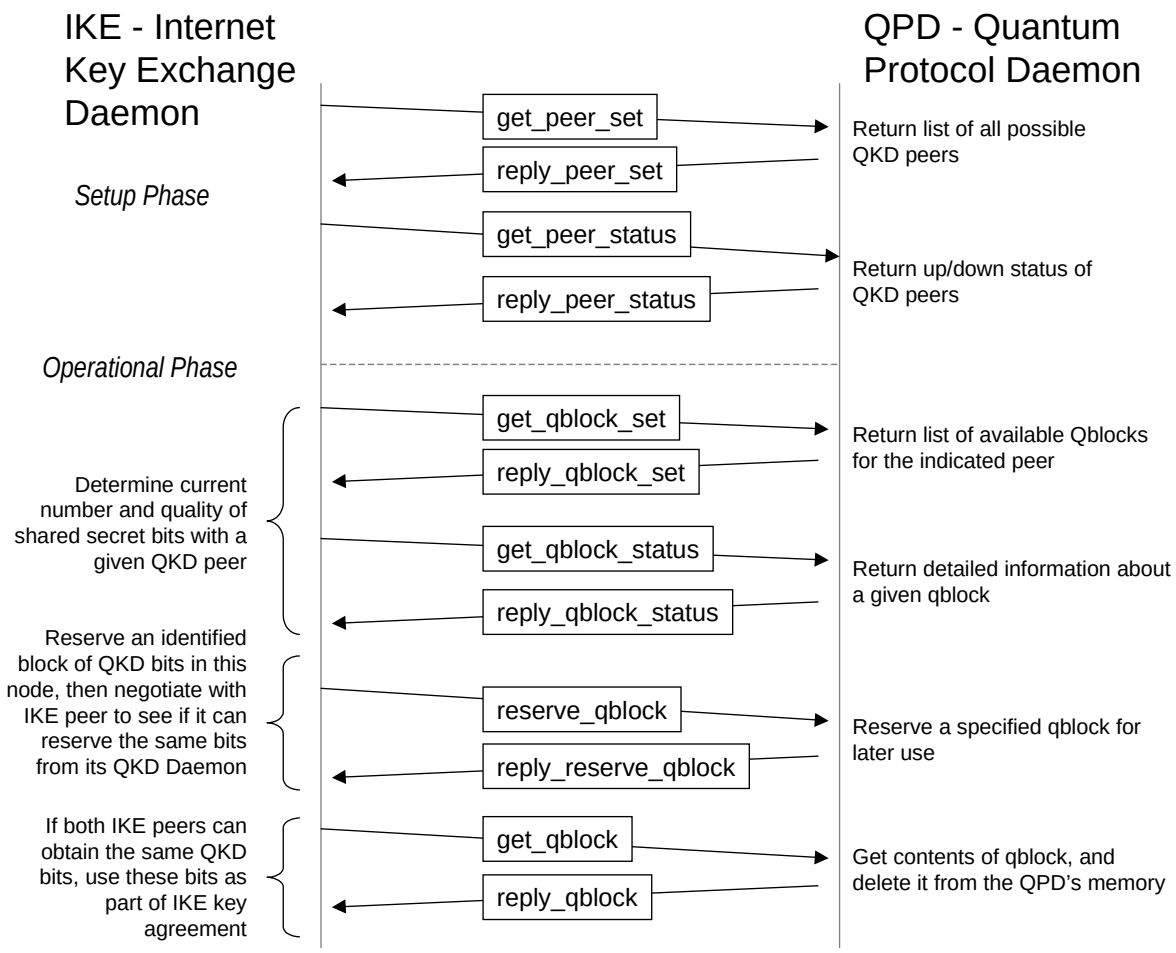


Figure 13-13. Overview of the IKD / QKD Interface.

As can be seen in the figure, this interface consists of a Setup Phase followed by an Operational Phase. During the Setup Phase, the IKE Daemon discovers the IP addresses for all potential QKD peers and requests the status for any peers that it cares about. Note that these IP addresses are the "black" IP addresses of the distant VPN computers.

During the Operational Phase, IKE assumes the active role and the QPD simply responds to its requests. Note that IKE may request how many Qblocks are present in the QPD and ready to be used as sources of shared secret bits, and the current status of any given Qblock.

IKE may further attempt to reserve one or more Qblocks, i.e., prevent them from being used for anything else. This reservation will be held valid by the QPD until the IKE Daemon actually gets those blocks for its own use, or releases the reservation so they can be used for something else. Such reservations are required in the negotiation between IKE peers, since they must come to a common agreement as to which sets of QKD secret bits they are going to both employ. The reservation approach allows one IKE Daemon to reserve a set of secret bits, tell its peer which bits it has reserved, learn from its peer whether the same set of bits are still available at that end of the link, and then "handshake" and agree to use those bits if they are available at both ends.

14 The BBN Key Relay Protocols

This section describes novel technology by which nodes in the Quantum Network may implement the "trusted" quantum key distribution network.

14.1 Overview of the BBN Key Relay Concept

When two VPN gateways do not have a point-to-point QKD link between them, but there is a path between them over QKD links through trusted routers, we need a way for them to agree on shared QKD bits. They do this by choosing a path (or multiple, independent paths) through the network, creating a new random Qblock, and essentially sending this Qblock one-time-pad encrypted across each link[34].

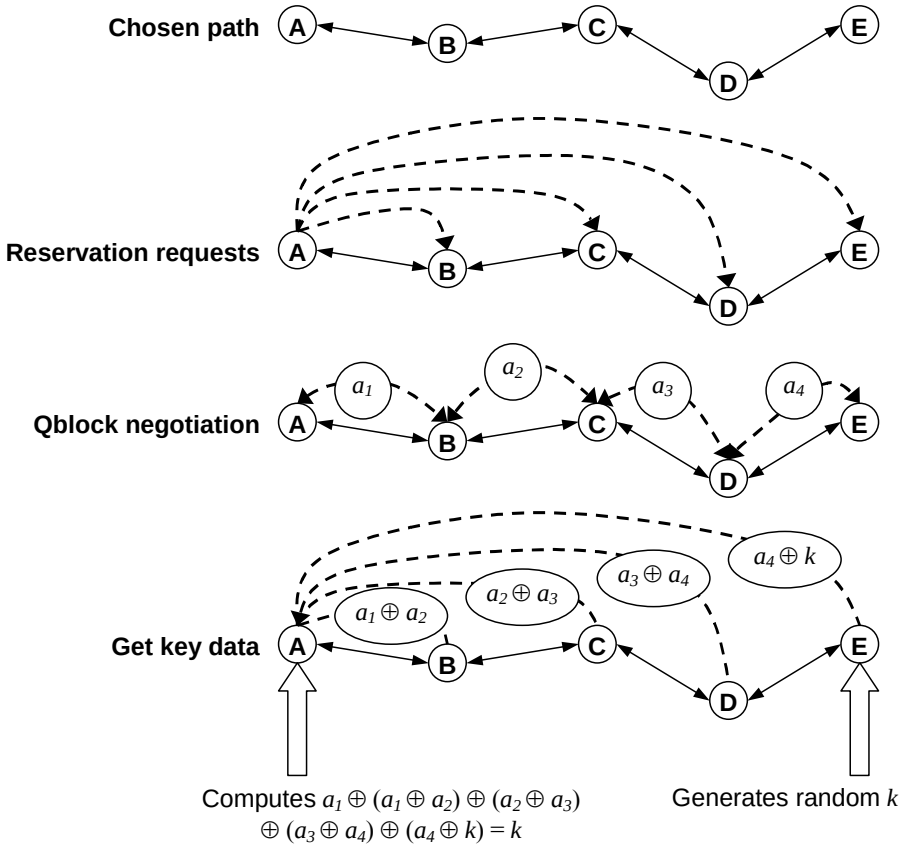


Figure 14-1. The BBN Key Relay Architecture.

[34] The actual protocol is slightly different, with the key material all passing from intermediate nodes to one of the endpoints of the path, but with the same underlying mathematics. This information flow makes an adversary's job very slightly harder (an adversary who has compromised one of the intermediate "trusted" nodes), requiring it to have also broken classical cryptography and to be able to eavesdrop on several different routes through the public network in order to discover the new Qblock.

The steps of key transport are illustrated in Figure 14-1. QKD Routing has selected a shortest path between nodes A and E[35]. Node A, who in this case initiates the key transfer, sends reservation requests to all other nodes along the path. Each of these other nodes then negotiates and reserves a Qblock with its predecessor along the path. All the intermediate nodes send node A the XOR of the Qblocks negotiated with the previous hop and the next hop, and node E, the destination, sends A the XOR of the previous-hop Qblock with the new, randomly generated Qblock. The final key, shared by A and E, is the new random block, denoted k in the figure.

The drawback of this scheme is that the secrecy of the key depends not just on the endpoints being trustworthy; the intermediate nodes must also be trustworthy. In order to make the network less vulnerable to the compromising of a single node, the key transport protocols can utilize multiple independent paths. In this case, the source node sends reservation requests along all the paths, and the destination node sends back the random Qblock XORed with all the previous-hop Qblocks. As before, the resulting Qblock shared by the source and destination will be the randomly-generated Qblock. An adversary would be unable to deduce the contents of this Qblock without compromising the source, the destination, or one intermediate node from each of the independent paths.

14.2 Auditing and Monitoring Material Derived from Key Relay

If an intermediate node is untrustworthy, it can compromise all keys that were transported through it. If the key was transported along two independent paths, a single compromised node will not endanger the secrecy of the key, but a pair of compromised nodes can if each sits along one of the independent paths. We assume that information about untrustworthy nodes may surface later, and in this case, it is important to know what keys may be endangered.

Thus one responsibility of the key transport software will be to keep a record of what nodes were involved in transporting each key block, and at what time it was transmitted, so that this information can be audited later, either to determine what other data may have been compromised by a hijacked router or, possibly, to make deductions about what routers may have been compromised if information has been leaked to an adversary.

14.3 Authenticating Material Derived from Key Relay

There are two levels of authentication used by the QKD protocols. Classical authentication is performed on all QPD – QPD communication, whether with neighbors or remote peers. Although classical authentication is all that we currently use, we anticipate adding universal-hash based authentication, with an information-theoretic guarantee against spoofing, to verify the identity of the endpoints of each QKD link, as well as ensuring the secrecy of the shared Qblocks produced on the link. Some aspects of universal-hash based authentication are pertinent to key relay.

In the presence of key relay and routing, the only way a transported key can become compromised is by one of the intermediate nodes being taken over, with its store of Qblocks and authentication bits still intact, by an adversary. In this case, using universal-hash authentication on the key transport messages from intermediate nodes would not add any assurance, while consuming authentication bits very quickly.

[35] QKD Routing is described in Section 16.

These messages will therefore not have universal-hash authentication. Similarly, forged routing update messages can only cause temporary denial-of-service, and so classical authentication is strong enough to protect against this.

In contrast, the final negotiation of the new Qblock(s) between the two endpoints *is* worth authenticating by universal hashing, because it establishes the identity of the endpoint even in cases when an intermediate node is compromised, and it consumes a small and bounded number of authentication bits to do so. Unfortunately, if there is an interloper along the route (or a conspiracy of them, if independent routes are used), they can still eventually spoof the identity of the endpoint by watching the transactions and deducing the secret keys until enough of the authentication pool has been replaced by compromised keys that they can guess the next authentication key bits.

14.4 The BBN Key Relay Protocol

This section provides a high-level introduction to the BBN Key Relay Protocol. Please consult the "QKD Protocol Design" Document for full details of this protocol.

Key relay begins with one node, the source, determining that it needs to create a key with a remote node, the destination. Of these two, the source is always the responder, i.e. the node with the higher node ID—only the responder will assign Qblock numbers between two nodes. The source node sends reservation requests to each intermediate node and the destination, indicating the predecessor in the path. Each node then negotiates with its predecessor a Qblock to use, which both it and the predecessor have reserved. It tells the source when this has been successfully completed. If all reservations were successful, the source then requests key data from all the intermediate nodes and the destination. The intermediate nodes send the XOR of the two Qblocks reserved with the previous hop and the next hop, and the destination sends the XOR of the previous hop Qblock with a new, randomly-generated Qblock. The new Qblock shared by the source and destination will be the random Qblock that the destination created. Figure 14-1 above illustrates this process.

If multiple independent paths are used, the destination sends the source the XOR of the random Qblock with the Qblocks for all the previous hops. The source XORs this with all the data received from the intermediate nodes and the first-hop Qblocks. In this way every Qblock except the one randomly generated by the destination is XORed twice, and so the result at the source is the new random block generated by the destination.

If any of the links is unable to supply a Qblock (perhaps because they were all reserved or consumed after the path was chosen), the intermediate node will send a failure message to the source, which will propagate the failure to all the other nodes along the path, releasing any reserved blocks. Failure can also occur if any of the nodes fails to reserve a Qblock within a configured timeout time.

15 QKD over Optical Switches

15.1 Background on Optical Switches

Toliver et al. have performed an initial set of experiments to measure the degradation in phase-modulated QKD incurred by optical switches[36]. They demonstrated successful transmission of QKD through three different types of optical switch elements. Insertion loss was the dominant effect on QKD throughput, ranging from 4.7 dB on a 2x1 opto-mechanical switch, 5.4 dB on a 2x2 $LiNbO_3$ switch, and 5.3 to 5.9 dB on a 4-port MEMS switch. At least one of these measurements included 10 km optical fiber along the path, which would incur 2 dB or more of this loss.

Thus Toliver's reported results appear to document insertion loss of perhaps 3 dB. This seems reasonable a priori. We note that our own 2x2 switch incurs a rather lower loss (1 dB).

15.2 The DARPA Quantum Network – Autonomous Optical Switches

BBN's work with optical switches is described in the following documents, which should be consulted for more detailed information:

- Photonics Subsystem Document: 2x2 Optical Switch

As of May 2004, the optically switched portion of the DARPA Quantum Network consists of two transmitters, Alice and Anna, and two receivers, Bob and Boris, interconnected through their key transmission link by a 2 x 2 optical switch, as depicted in Figure 15-1. In this configuration, either transmitter can directly negotiate a mutual key with either receiver. The switch is optically passive so that the quantum state of the photons that encode key bits is not disturbed.

At present, the switch is controlled by a direct line from Alice's OPC. Special purpose software on this OPC sets the switch to the BAR setting for an interval, then to the CROSS setting for the same interval, back to the BAR setting, and so forth.

That is, the switch control is currently completely decoupled from the operation of the QKD links that traverse the switch. At present, the QKD receivers (Bob and Boris) simply notice that they are receiving Raw Qframes from a new source and react accordingly.

[36] Paul Toliver, Robert J. Runser, Thomas E. Chapuran, Janet L. Jackel, Thomas C. Banwell, Matthew S. Goodman, Richard. J. Hughes, Charles G. Peterson, Derek Derkacs, Jane E. Nordholt, Linden Mercer, Scott McNown, Art Goldman, and John Blake, "Experimental Investigation of Quantum Key Distribution Through Transparent Optical Switch Elements," IEEE Photonics Technology Letters, v. 15, n. 11, pp. 1669-1671, November 2003.

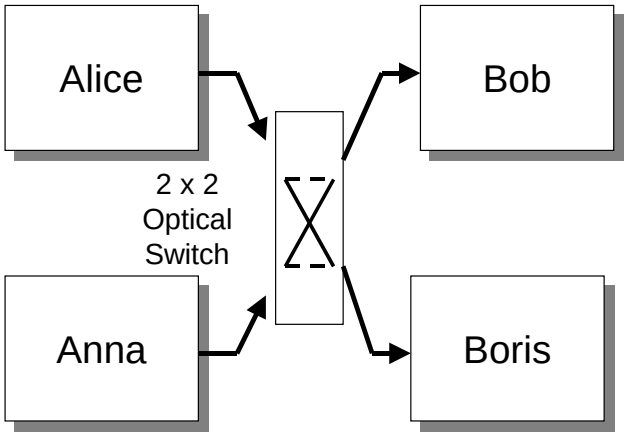


Figure 15-1. Optical Switch Interconnecting Two Mark 2 Weak-Coherent Systems.

The switch chosen for this network is a standard telecommunications facilities switch that operates by moving reflective elements that change the internal light path to produce either a BAR or CROSS connection. It is operated by applying a TTL-level pulse to either the BAR or CROSS pin for 20 ms and latches in the activated position. Switching time is 8 ms and optical loss is <1 dB.

Figure 15-2 shows a photograph of the switch mounted on a PC board with the electrical interconnects on the left and optical interconnects on the right.

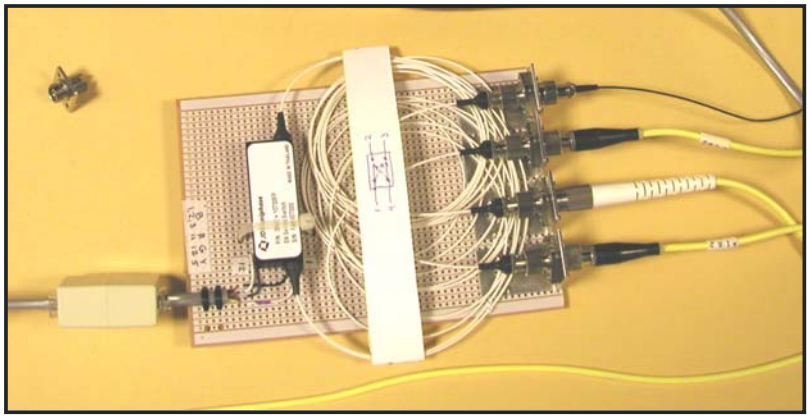


Figure 15-2. The 2x2 optical switch mounted on a PC board.

15.3 The DARPA Quantum Network – Optical Switching Protocols

This section describes a suite of novel protocols and algorithms that enable the "untrusted" version of the DARPA Quantum Network. These protocols and algorithms allow QKD endpoints to set up, monitor, and tear down virtual circuits for QKD photons through a series of one or more passive optical switches.

16 The BBN QKD Routing Protocols

This section briefly introduces the BBN-designed QKD Routing Protocols in the DARPA Quantum Network. At present, two distinct protocols are envisioned:

- Routing Protocols for QKD Key Relay
- Eavesdropping-Aware Routing

Please refer to the QKD Protocol Design Document for further information about these protocols.

16.1 BBN Routing Protocols for QKD Key Relay

Key routing is responsible for choosing the path or paths used by key relay. (Please see Section 14, "The BBN Key Relay Protocols," for information about key relay.)

Each node will maintain a database of the full link state of the network. For each node in the network it will keep the node's ID and a list of neighbors. For each neighbor, it will keep the neighbor's node ID and the current metric of the link between them. When finding a single path between two nodes, the path with the smallest total metric (i.e. the smallest sum of the link metrics along the path) will be used. Dijkstra's algorithm can be used for finding this path. When finding multiple independent paths, the set of independent paths with the smallest total metric will be used (see [B97] and [B99] for a generalization of Dijkstra's algorithm for multiple independent paths).

To keep the link-state databases up-to-date, a single link-state advertisement message is used:

> **`ROUT1_LSA`**: this message carries in its data the node ID of the sending node, then for each neighbor the node ID of the neighbor followed by the link metric. The metrics are 32-bit integers in network byte order.

Each node should send one **`ROUT1_LSA`** message to every other node in the network every LSA update interval, a configurable parameter set to one minute by default. Each LSA update is an independent job, with its own job number. The LSA both starts and ends the job.

The link metric used by routing is:

$$m = \begin{cases} 100 + \dfrac{1000}{q - t}, & q > t \\ \infty, & q \le t \end{cases}$$

where m is the link metric, q is the number of Qblocks expected to be available on the link (and not reserved) in one LSA update interval, and t is a configurable threshold (default value 5) for the minimum number of qblocks to leave on an active link. The current estimate of the number of available Qblocks one update interval from now is simply the number available now, but in the future a predictive model may be used.

The routing protocol will still work if some nodes use different metrics, so it would be legal for a node to, for example, put a higher threshold on certain links to guarantee that they have more Qblocks available.

16.2 BBN Routing Protocols for Eavesdropping-Aware Routing

This section describes novel technology by which nodes in the Quantum Network become aware of a too-high level of noise on the quantum key distribution link, which may indicate the presence of eavesdropping, and how they then "route around" these eavesdropped links.

17 The IPsec Protocol Suite

This section provides a basic introduction to the role of IPsec and the Internet Key Exchange (IKE) protocol suite in offering secure communications through an untrusted Internet, and describes our extensions to IKE that marry quantum cryptography into the Internet's overall security architecture.

17.1 Defining Documents and Standards Status for IPsec and IKE

IPsec is an architectural framework for secure communications within the Internet Protocol suite. This framework is defined by a standards-track document within the Internet Engineering Task Force (IETF), which acts as the worldwide standards organization for Internet protocols:

RFC 2401 Security Architecture for the Internet Protocol

The Internet Key Exchange (IKE) protocol is one component within the IPsec architecture. Its role is to permit two endpoints to agree first on which cryptographic protocols and algorithms they wish to employ for a given security association, and second on the keys they use to encrypt and/or authenticate subsequent message traffic within that security association. IKE is defined by its own standards-track document within the IETF:

RFC 2409 The Internet Key Exchange (IKE)

At time of writing (June 2002), the Internet community has mixed feelings about IKE's suitability as a standard, based on perceived technical shortcomings in the protocol. There is a significant chance that IKE will be rejected as a standard. If so, this will leave an awkward hole in the IPsec protocol suite since key-agreement is a necessary tool. The Internet community is currently considering a newer protocol for key agreement, termed Just Fast Keying (JFK), but it is too early to tell whether this new protocol will indeed become widespread, or indeed, even if IKE will be rejected.

The DARPA Quantum Network needs *some* IP-level key agreement protocol, and we strongly prefer to base our work on a widely deployed – and if possible, standards-track – protocol. However, the internal architecture of the DARPA Quantum Network is quite flexible about exactly which key agreement protocol is employed. Indeed, it can support the use of multiple key agreement protocols running in parallel. Since IKE is at present (January 2002) the most widely deployed Internet key agreement protocol, we will continue to employ IKE as the basis for our quantum cryptographic extensions. We will continue to monitor the Internet market, however, and shift away from IKE and towards its replacement, should IKE indeed become irrelevant.

17.2 Basic Concepts for IKE

Although IKE is a relatively complicated protocol, its basic concepts are straightforward. Figure 17-1 below depicts the most important elements involved in an ongoing relationship between two IKE peers. This illustration is intended to be high-level and schematic rather than a detailed depiction of the actual software architecture that implements IKE.

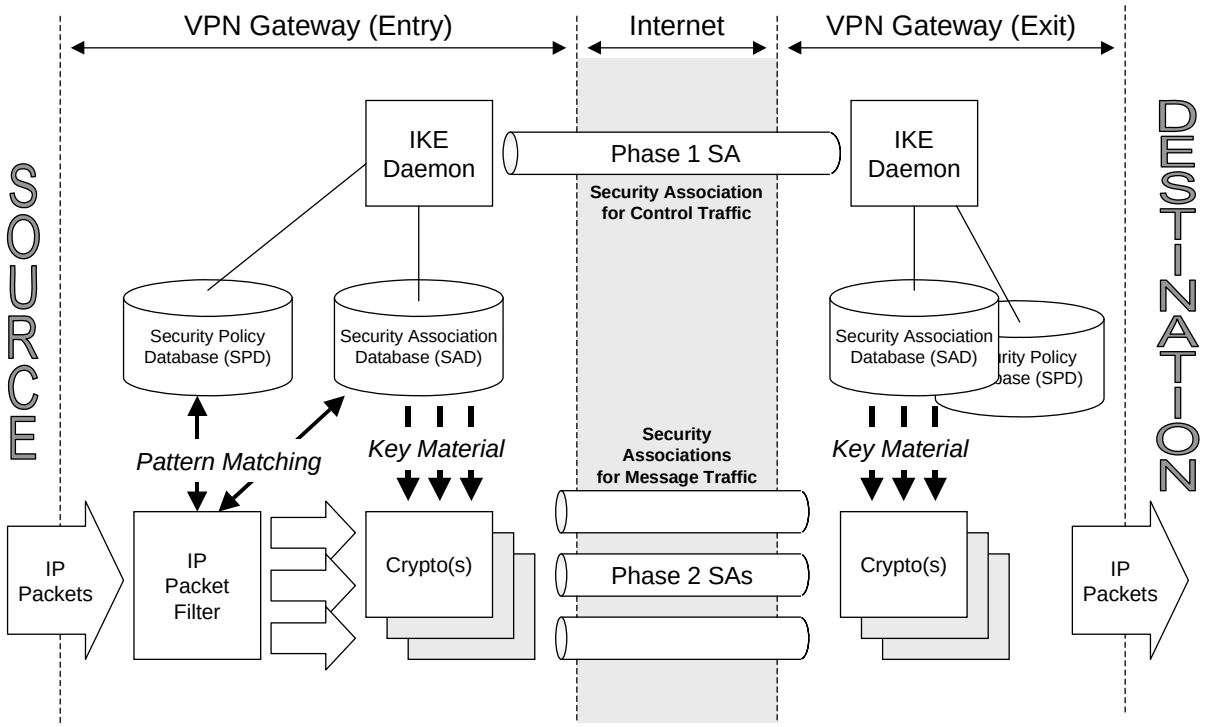


Figure 17-1. Major Components in IKE and Virtual Private Network (VPN) Gateways.

This diagram shows a "source" private enclave at the left of the picture and a "destination" at the right, with data traffic (IP datagrams containing message traffic that must be protected) flowing from right to left. In point of fact, practical setups are inevitably symmetric so that traffic can flow in both directions but we've shown it as asymmetric to better bring out the functions of the various components as traffic flows from left to right.

Each VPN Gateway contains a control plane and a data path. The control plane is used for setting up, administering, and tearing down "security associations" (encrypted tunnels) for the data traffic as needed. All IKE interactions take place in the control plane between a pair of peer IKE Daemons.

All message traffic, by contrast, follows the data path. As a message datagram, i.e. an IP packet, enters the VPN Gateway at the left, it is first passed through a packet filter which determines exactly how this packet should be treated. Assuming that it should be forwarded through a security association, it is then passed through the appropriate encryption device or algorithm (crypto) in order to encrypt it, add authentication fields, add integrity checks, and so forth, as are required for that particular security association. The secured packet then transits the public (presumed hostile) Internet to the exit gateway, where it is decrypted, integrity-checked, and so forth. Assuming it passes all tests, the decrypted datagram is then forwarded on in the clear into the "destination" private enclave.

As shown in Figure 17-1, two databases play a key role in this process. When a packet is filtered, its address tuple[37] is first checked against the Security Association Database (SAD), which maintains

[37] The address tuple consists of some or all of the following fields drawn from an IP datagram: source IP address with mask, destination IP address with mask, protocol, source port, destination port.

essential information about all currently-existing security associations (SAs). If a match is found in this database, the packet is then handled according to the rules for the matching security association, as recorded in the SAD.

If the packet does not match any current security association, however, its tuple is then checked against the Security Policy Database (SPD) to see if a new security association should be created for this datagram. If so, a great deal of work must take place on the control plane in order to set up this new security association. This work is performed by the IKE Daemon, and is often called "setting up a Phase 2 security association" for reasons that will become evident below. Once this new security association has been set up at both peer gateways – including the fact that synchronized key material has been positioned in both devices – the IKE Daemon then adds a new entry to the SAD reflecting this new association, and the datagram can be processed as described above.

Note that the two IKE Daemons must communicate with each other in order to establish security associations for data traffic. Needless to say, their communication is very sensitive and must be protected. Therefore all such IKE control traffic is carried via IP datagrams through a special IKE-to-IKE security association.

It should be evident that the IKE-to-IKE security association must be active before any traffic SAs can be created, and as a consequence it is created before any of the other SAs. As a result, in IKE jargon it is generally termed a "Phase 1 SA." Note that there may be multiple Phase 1 SAs active within a single IKE Daemon, because it needs one for each peer IKE Daemon with which it will communicate. Given this terminology, the traffic-carrying SAs are naturally termed "Phase 2 SAs."

Element in Figure 17-1	Description
IKE Daemon	The software entity that implements our augmented version of the Internet Key Exchange (IKE) protocols and algorithms. (BBN modification of the IKE Daemon supplied by KAME.)
Security Policy Database (SPD)	A database together with algorithms that classify IP datagrams to determine which datagrams belong in which security associations. This is done by pattern-matching of various fields in the IP datagrams with rule sets in the database. If a datagram is found that requires a security association but does not yet have one in place, a signal is sent from the SPD to the IKE Daemon requesting that such an association be established.
Security Association Database (SAD)	A database together with algorithms that perform IPsec actions on IP datagrams as needed for a given security association, e.g., encryption or decryption, authentication, encapsulation, and the like.
IP Packets	Datagram packets conforming to the standard Internet Protocol (IP) suite. These may be either the older or new versions of the IP suite, that is, either IPv4 or IPv6.
IP Packet Filter	An IP packet filter inspects all packets in transit and performs pattern matching on them, as described below.
Crypto(s)	This is where the "crypto" is implemented in IPsec. These cryptos may be implemented in software, hardware, or a combination of the two. The crypto may be responsible for additional operations beyond classic

	encryption and decryption, e.g., it may provide cryptographic hashes.
Pattern Matching	Pattern-matching is the process of checking an incoming IP datagram from a private enclave ("red" datagrams) against those lists of address tuples maintained as indexes to the SAD and SPD. If a pattern is matched, the datagram will be handled according to the corresponding rules for that pattern. Otherwise it is handled in whatever way non-matching datagrams are handled in the system, e.g., discarded, sent in the clear, etc.
Key Material	
Internet	This is an "untrusted" Internet. It may be the public Internet or alternatively it may be a private network based on Internet technology that is not fully trusted by the communicating enclaves.
Phase 1 SA	A security association between peer IKE Daemons that carries IKE control traffic between the two VPN gateways.
Phase 2 SA	A security association between two VPN gateways that carries message traffic for a particular Virtual Private Network traffic flow.

As a matter of terminology, we note that an IPsec SA is *simplex* or one-way. For instance, an SA may be formed from Alice to Bob, but is not inherently bi-directional. Thus IPsec generally works in terms of a so-called "SA bundle" that consists of two SAs – one in each direction.

17.3 Authentication in IPsec / IKE

Authentication is the process performed by Alice to determine that she is really communicating with Bob, and not some other entity such as Eve performing a man-in-the-middle attack, and for Bob to determine that he is really communicating with Alice. In general, the IPsec / IKE architecture may perform authentication of three different types of interactions between Alice and Bob:

- When creating a Phase 1 SA between themselves as IKE peers, which happens at startup and also periodically as the Phase 1 SAs roll over
- When exchanging control traffic (through the Phase 1 SA) in order to create a Phase 2 SA between themselves for carrying user message traffic
- On individual message traffic packets themselves, to ensure that each message packet is indeed from the authorized IPsec peer

IKE authentication for Phase 1 SAs works as follows. First, the two would-be IKE peers exchange random nonces, cookies, and half-keys for a Diffie-Hellman key exchange. Then they create a secret session key based on these random inputs and the Diffie-Hellman algorithm. They then perform Phase 1 authentication by exchanging messages encrypted with this session key.

IKE provides a choice of several authentication mechanisms for Phase 1 SA establishment. We employ RSA public-key signatures and X.509 certificates, where each party signs a hash that includes g^{xy} and the nonces, and authenticates the other party by checking its signature. It is currently an open research question as to whether we would like to employ secret keys, e.g. as disseminated by QKD protocols, instead of or in addition to public key mechanisms. We note the IKE does support authentication based on

pre-shared keys which seems like a potentially good mechanism to marry with continuous authentication based in part of QKD techniques.

The establishment of a Phase 1 SA between Alice and Bob gives rise to a shared set of secret bits that are conventionally termed the SKEYID. Both entities then divide this set of bits into three distinct subsets: SKEYID_a for authentication of control traffic in the Phase 1 SA, SKEYID_e for encryption of control traffic in the Phase 1 SA, and SKEYID_d which is used as partial input when creating Phase 2 SA keys (and hence ultimately for protecting message traffic through a given Phase 2 SA).

Note that this SKEYID material is thus used at some rate, for protecting both Phase 1 control traffic and for protecting user message traffic. It is thus important that this raw key material be "rolled over" at a fast enough rate so that its reuse doesn't compromise system security. That is, Alice and Bob must re-agree on SKEYID material from time to time in order to refresh these secret bits. They do so by rolling over from an older Phase 1 SA to a newer one at fair frequent intervals, as described in the next section. Whenever a new Phase 1 SA is activated, it will bring with it a new SKEYID which is then used for subsequent transactions until it in turn is replaced.

17.4 Keys and Key Rollover in IPsec / IKE

Figure 17-2 below presents a data flow diagram that shows the various kinds of key material used in IPsec / IKE and how this material is derived. It is annotated with the various places at which cryptographic random numbers are introduced. Shaded stars tag those quantities based on a local source of random numbers; non-shaded stars tag those quantities that are based on a peer's random number source. The little gray box attached to the side of a "crypto" indicates the key input for that box; all other inputs should be concatenated and taken as message text input. For clarity, we have presented this diagram from the initiator's point of view.

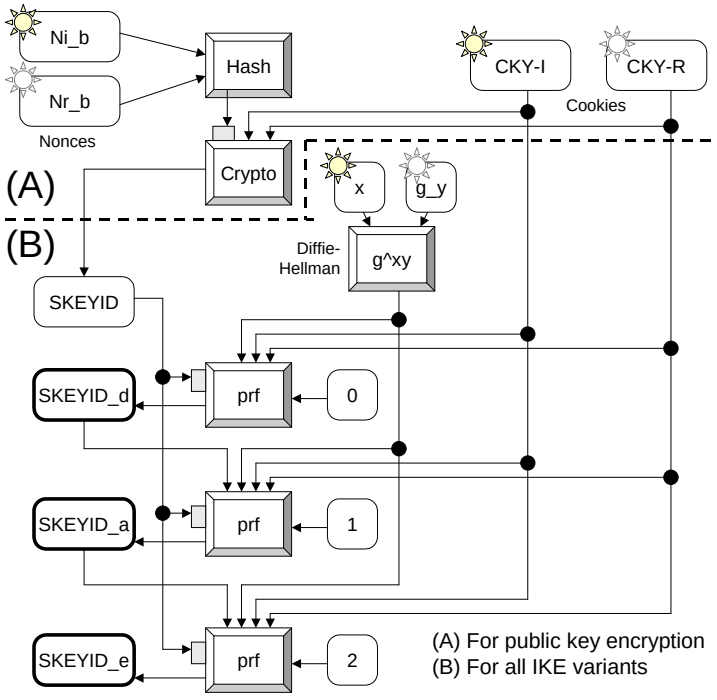


Figure 17-2. Data Flow Diagram for IKE Key Material, Public Key Variant.

For convenience, the actual defining text from the IKE standards document (RFC 2409) is reproduced below. We have echoed this IKE terminology in **Figure 17-2. Data Flow Diagram for IKE Key Material, Public Key Variant.**.

```
CKY-I and CKY-R are the Initiator's cookie and the Responder's
cookie, respectively, from the ISAKMP header.

Nx is the nonce payload; x can be: i or r for the ISAKMP initiator
and responder respectively. (Note that Ni_b and Nr_B are the bodies
of the nonce payloads, i.e., the nonces themselves.)

g^xi and g^xr are the Diffie-Hellman ([DH]) public values of the
initiator and responder respectively.

g^xy is the Diffie-Hellman shared secret.

prf(key, msg) is the keyed pseudo-random function-- often a keyed
hash function-- used to generate a deterministic output that
appears pseudo-random.  prf's are used both for key derivations and
for authentication (i.e. as a keyed MAC). (See [KBC96]).

|  signifies concatenation of information-- e.g. X | Y is the
concatentation of X with Y.

For signatures:            SKEYID = prf(Ni_b | Nr_b, g^xy)
For public key encryption: SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)
For pre-shared keys:       SKEYID = prf(pre-shared-key, Ni_b | Nr_b)

SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2)
```

The actual key size needed for some encryption or hash algorithm depends on the specific algorithm being employed. Some such algorithms always use the same key size. For others, which can accept a range of key sizes, the Security Policy Database (SPD) must list the actual key size that should be employed for a given SA.

Every IKE security association has a maximum lifetime which governs how long the key material for that association can be used. This lifetime can be expressed either in time (seconds) or in data encrypted (kilobytes) and is configured via the Security Policy Database (SPD) entry for a given security association. After the lifetime expires, a new security association must be negotiated and it will bring with

it fresh key material. This is sometimes termed "key rollover," because it replaces an older key by a newer one while still protecting the same underlying traffic flow.

IKE includes mechanism to enable "soft rollovers" of security associations, so that a new association can be formed before the old one has been totally torn down. This feature helps eliminate cases in which the underlying traffic flow is disrupted because of lack of matching cryptographic material at both the entrance and exit gateway for that traffic flow. To this end, IKE implements "soft" lifetimes for security associations, i.e., those points at which a replacement SA should be negotiated and a rollover should begin, and "hard" lifetimes at which the old security association must be deleted whether or not a new one is present to replace it.

Conventional IKE has a crisply defined concept of Perfect Forward Secrecy (PFS). To quote RFC 2409, "Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key. For PFS to exist the key used to protect transmission of data MUST NOT be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys."

Finally, we note in passing that the IKE standard requires use of Diffie-Hellman key exchange. We may augment this, if we wish, with a further source of secret shared bits obtained by quantum cryptography, but one cannot delete the Diffie-Hellman exchange without violating the standards definition.

17.5 Timeouts and Corrupted Keys in IPsec / IKE

In this section, we would like to call special attention to a few rarely-exercised parts of the IKE design that may have some impact on the DARPA Quantum Network's overall system design.

The first such aspect concerns *timeouts*, in particular, the maximal amount of time that can elapse during an IKE negotiation. Such values are often set to 10s of seconds for Phase 1 negotiation, and less than 10 seconds for Phase 2. These values may be too small for systems employing QKD since it may take a while to accumulate enough bits for a successful negotiation. In addition, of course, this narrow window makes Eve's denial-of-service attacks somewhat easier since she must block IKE messages during only a relatively short time in order to bring down the security association(s).

The second concerns what IKE does when Alice and Bob believe they posses secret bits in common but in fact these two sets of bits are not identical. This may well happen in quantum cryptography, since noise on the single-photon channel can only be detected and corrected probabilistically. See Section 13.8 for a discussion of QKD error correction. As it happens, IKE has no mechanisms for noticing or dealing with such cases. The result appears to be that all security associations that employ key bits derived from this corrupted information will fail to properly encrypt / decrypt traffic. This situation will apparently continue until the security association is renewed, i.e., rolls over to a new security association.

17.6 Overview of IKE Extensions for Quantum Cryptography

Our first extensions fall into two basic categories: those that use QKD techniques for agreement on secret keys that are then employed as seeds for conventional symmetric ciphers (e.g. AES and 3DES) with continual and automatic reseeding by fresh QKD bits, and those that use a sequence of QKD bits as a one-time pad or Vernam cipher for the message traffic. We briefly note each extension in this section.

- In the rapid-reseeding extension, the central change is to include reconciled QKD bits as input to the IKE Phase 2 hash, so that keys protecting IPsec Security Associations (SAs) are derived from QKD.

- In the one-time pad extension, the central change is to introduce one-time pad concepts into IKE and IPsec traffic processing, with all the adjutant changes to protocols and algorithms required, and the necessary linkages into the secret bits obtained via QKD techniques.

Ancillary changes include policy mechanisms to specify when either of these extensions should be used, on a per-tunnel basis, and negotiation mechanisms to agree on which QKD bits will be used.

17.7 IKE Phase 1 Extensions for QKD

We do not currently plan any IKE Phase 1 extensions in Year 1. Thus as currently planned, the DARPA Quantum Network will not employ any QKD techniques in order to (a) authenticate IKE peers, or (b) protect the control traffic exchanged between IKE peers.

We note that Phase 1 SAs are used for essentially only one purpose: to protect key management messages for Phase 2 SAs. Since there is nothing in the Phase 2 control traffic that requires the level of protection provided by QKD techniques, it appears that Phase 1 SAs do not need any special level of protection.

17.8 IKE Phase 2 Extensions for QKD

Standard IKE Phase 2 currently supports optional PFS. We have added an option for Quantum Perfect Forward Secrecy (QPFS) that functions in the same structural way, but uses QKD bits rather than a fresh DH exchange. By Phase 2 PFS, we mean the double-ephemeral DH exchange in Phase 2, used only for the Phase 2 Security Association (SA) being negotiated.

In our first release of the DARPA Quantum Network, IKE Phase 2 has an additional option, "Phase 2 QPFS." This adds QKD bits to the hash that determines the Phase 2 keying material. The choices to use Phase 2 PFS and Phase 2 QPFS are independent; either or both a conventional DH value or QKD bits can be included in the hash. The key design issues in this area are as follows:

- What degree of QKD processing is required
- How the use of the option is controlled (by policy) on each side,
- How the negotiation is done within IKE,
- How the number of bits to be used is chosen,
- How the specific bits to be used are specified, and
- How the bits are used in the hash to produce session keys.

Please refer to the "IPsec Quantum Cryptography Extensions" design document for a detailed discussion of this issues, along with a specification of the resulting protocol, algorithm, and database modifications used in the DARPA Quantum Network. In addition, we have implemented an interface between the IKE Daemon and the Quantum Protocol Daemon for reservation of and access to secret bits obtained by quantum cryptography. Please see the "IKE / QKD Interface Control Document" for a specification of this interface.

Figure 17-3 and Figure 17-4 show our extended versions of IKE's Security Policy Database (SPD) and Security Association Database (SAD) in highly simplified form. Note that new fields have been added to support VPNs protected by quantum cryptography, which may be disabled, allowed, or required on a per-tunnel basis in the DARPA Quantum Network.

Index	Peer IKE IP Addr.	PFS Req'd? (*)	QPFS Req'd? (*)	
Tuple #1	Peer Addr #1	No / Permit / Require	No / Permit / Require	
Tuple #2	Peer Addr #2	No / Permit / Require	No / Permit / Require	
(etc)	*(*) indicates a new field added to support quantum cryptography.*			

Figure 17-3. Simplified Diagram of Security Policy Database (SPD) with QKD Extensions.

Index	Peer IKE IP Addr.	PFS In Use? (*)	QPFS In Use? (*)	
Tuple #1	Peer Addr #1	Yes / No	Yes / No	
Tuple #2	Peer Addr #2	Yes / No	Yes / No	
(etc)	*(*) indicates a new field added to support quantum cryptography.*			

Figure 17-4. Simplified Diagram of Security Association Database (SAD) with QKD Extensions.

18 Analyses of System Performance

This section provides analysis of the overall system performance, based on a series of lower-level analyses of critical subsystems' performance.

At present, this analysis is very preliminary. It is still work in progress and has not yet been checked against any measured results. Thus it is included here mainly as a place-holder for the better-validated results that will be forthcoming during late 2002 and into 2003.

18.1 Simplified Link Budget for the Mark 2 Weak-Coherent Link

Table 18-1 below presents a simplified link budget for the Mark 2 Weak-Coherent Link, with an explanation for the loss at each stage of the process[38].

Table 18-1. Simplified Link Budget for Mark 2 Weak-Coherent Link.

Item	Rate / Loss
Max detection rate for Current InGaAs detectors	< 1 MHz
Mean Photon Number approx. 0.1 photons / pulse	10 db
Fiber channel optical loss	10 db
Optical Receiver Loss for Phase Modulated Pulses	3 db
Quantum Efficiency for Current InGaAs detector	10 db
Basis guessing mis-match	3 db
Information Revealed in Error Detection and Correction	1 db
TOTAL LOSSES	~ 37 db
Delivery Rate for Privacy-Amplified Key Material	~ 200 bits/second

18.2 Analytic Model of the Mark 2 Weak Coherent System

This section presents the current version of the Matlab / Octave model we have developed in order to analyze the expected efficiency of current and projected fiber-based QKD systems in the DARPA Quantum Network. The complete model is provided in this section. Some aspects of the model have been derived from the QKD literature, but most have been developed from first principles. Dr. John Myers of Harvard University has provided many of the equations in this model; the authors have provided the remainder. Of course the authors are solely responsible for any flaws in the model itself.

This model provides for a wide range of input parameters such as pulse rate, mean photon number at Alice, attenuation, detector efficiency, dark count, and after-pulsing characteristics, residual phase error in the Mach-Zehnder interferometers, and so forth. It also provides input parameters for higher layers of the QKD protocol stack, such as the sifting protocol employed, information revealed during error detection and correction, entropy estimation technique, etc.

[38] This simplified link budget was developed in discussions with Science Research Laboratory, Somerville, MA.

We briefly discuss these inputs, and the associated calculations, in the following paragraphs. Although the model provides basic estimates for a range of physical and protocol phenomena, it is by no means complete. For example, it does not include any characterization of stray light, of chromatic or polarization mode dispersion, and so forth. However, the current version of this model has been validated against our QKD systems running both through a fiber spool in the laboratory and through a 17km fiber loop between BBN and Harvard University, and its results agree well with experimental measurement. Thus it appears to capture at least the most important drivers for realistic system behavior.

As shown, the model inputs represent a fiber-based system with a 5 Mb/s pulse rate, 0.1 mean photon number (μ), operating through 10.55 km of telecommunications fiber with an overall fiber attenuation of 2.5 dB. The average receiver loss factor is 10.4 dB, with a residual phase error in the Mach-Zehnder interferometers of 3 degrees after both passive and active path length stabilization. The path length stabilization and framing overhead results in a duty cycle of 80% for usable QKD bits. Detector efficiency is 13%, with mis-steered light occurring in 0.9% of the detections, and a dark count probability of 2.8×10^{-5} per pulse. At higher layers of the QKD protocol stack, the traditional BB84 sifting algorithm is modeled, with the BBN variant of the Cascade error detection and correction protocol using a block size of 4,096 bits with 64 sets, the traditional BBBSS92 entropy estimate, and a residual confidence level (the probability that Eve has more information than estimated) of 10^{-6}. These values capture the state of our Mark 2 Weak Coherent QKD systems as of January 2004.

It should be apparent from inspection of the model that these parameters can be readily adjusted to model other fiber-based systems, e.g., different detector characteristics, protocol behavior, and so forth. One could also extend the model to free-space systems, or systems based on pairs of entangled photons, but this would require that additional equations be added to the model rather than mere adjustment of input parameters.

```
% File: model.m
%
% Description: Analytic model of QKD throughput
%
% $Id: model.m,v 1.2 2004/05/21 20:24:04 dpearson Exp $
%
% Copyright (c) 2004 by BBN Technologies
%
% This is a Matlab / Octave model of the QKD throughput of a weak-coherent
% source using BB84 through fiber, given various parameters of the system.
% Key parameters include:
%
%       pulseRate       -- the repetition rate of the source, in Hz
%       dutyCycle       -- portion of pulses for payload (vs. header, training)
%       mpn             -- the mean photon number per pulse at Alice
%       fiberLength     -- the length of fiber, in km
%       fiberLoss       -- the attenuation of the fiber, in dB/km
%       rxLoss          -- receiver loss, in dB (Myers eta_rec, as dB)
%       detEff{0,1}     -- detector efficiency, for each detector (eta_det)
%       detLeak{0,1}    -- leakage of other's light into this detector (epsilon)
%       pDark{0,1}      -- probability that detector fires w/ no light
%       pAfter{0,1}     -- probability pulse results in unsuppressed afterpulse
%       residPhase      -- residual phase error, RMS, in radians
%       strayPh         -- stray light entering receiver, in photons per gate
%
%       blockSize       -- number of bits in block for EDAC / privacy amplify
%       nEdacSets       -- number of subsets for EDAC
%       estType         -- entropy estimate type ('Bennett', 'Slutsky', 'Myers')
```

```
%       confidence      -- probability Eve has more information than estimated
%       siftType        -- type of sifting ('BB84', 'SARG')
%
%       eveChan         -- for PNS, Eve's multiplier on fiberLoss (0=perfect)
%
% This file defines typical values for these variables (which are all
% global variables), and functions which use them to compute the rate
% of detects, errors, and finished bits.  To try different scenarios,
% you can simply modify the global parameters and re-execute the function.

global pulseRate dutyCycle mpn fiberLength fiberLoss rxLoss residPhase strayPh
global detEff0 detEff1 detLeak0 detLeak1 pDark0 pDark1 pAfter0 pAfter1
global blockSize nEdacSets estType confidence siftType eveChan

pulseRate = 5e6;                % Alice-Bob link runs at 5MHz
dutyCycle = .8;                 % Measured duty cycle
mpn = .1;                       % Target value (was calibrated recently)
fiberLength = 10.55;            % Length of fiber spool, in km
fiberLoss = .237;               % dB/km for spool, if total = 2.5dB
rxLoss = 10.4;                  % measured loss (dB, average over all paths)
residPhase = 3 * pi/180;        % not measured recently
detEff0 = .117;                 % from analysis of data
detEff1 = detEff0;
detLeak0 = .009;
detLeak1 = detLeak0;
pDark0 = 2.8e-5;
pDark1 = pDark0;
pAfter0 = .001;                 % SW/HW suppression should keep this quite low
pAfter1 = pAfter0;
strayPh = 0;                    % negligible in lab

blockSize = 4096;               % Configured min (average slightly higher)
nEdacSets = 64;                 % Configured
estType = 'Bennett';            % Configured
confidence = 1e-6;              % Hard-wired
siftType = 'BB84';              % Configured

eveChan = 0;                    % Assume Eve has perfect fiber

% sourceRate -- the raw rate of symbols at the source (not counting
% attenuation)

function r = sourceRate
    global pulseRate dutyCycle
    r = pulseRate * dutyCycle;
endfunction

% Utility function to compute the probability of the union of a number of
% independent events

function p = probOr(varargin)
    p = 1;
    for i = (1:nargin)
        p = p * (1-varargin{i});
    endfor
    p = 1 - p;
endfunction

% Here we estimate the probability of the different kinds of detections, and
% turn those probabilities into the sifted rate and QBER.
%
% pmCorr = probability that correct detector fires when bases match
% pmIncorr = probability that incorrect detector fires when bases match
% pwDetect = prob that detector fires when bases wrong (same for both D0 & D1)

function [rate, qber] = siftedRate
    global mpn fiberLength fiberLoss rxLoss residPhase strayPh
```

```
    global detEff0 detEff1 detLeak0 detLeak1 pDark0 pDark1 pAfter0 pAfter1
    pDark = (pDark0 + pDark1) / 2;
    pDark = probOr (pDark, 1-exp(-strayPh*(detEff0+detEff1)/2*.1^(.1*rxLoss)));
    e = (detEff0*detLeak0 + detEff1*detLeak1) / (detEff0 + detEff1);
    atten = .1^(.1*(fiberLength*fiberLoss + rxLoss));
    c = (detEff0+detEff1)/2 * mpn * atten / (1+detLeak0+detLeak1);
    pwDetect = probOr (pDark, 1-exp(-c*(e + .5)));
    pAfter = pwDetect * (pAfter0 + pAfter1) / 2;
    pwDetect = probOr (pwDetect, pAfter);
    pmCorr = probOr (pDark, pAfter, 1-exp(-c*(e + cos(residPhase/2)^2)));
    pmIncorr = probOr (pDark, pAfter, 1-exp(-c*(e + sin(residPhase/2)^2)));
    pmValid = probOr (pmCorr, pmIncorr);
    rate = pmValid / 2 * sourceRate;
    qber = (pmIncorr - pmCorr*pmIncorr) / pmValid;
endfunction

% EDAC overhead -- this is for the amount of extra information revealed,
% per bit, given the error rate.  This is specifically for the BBN variant of
% Cascade, other protocols are likely to differ slightly.  This also
% represents an average, over many blocks of slightly varying size and
% error rate.  The estimate does not include the error bits themselves.

function ovhd = EDACoverhead (qber)
    global nEdacSets blockSize
    ovhd = qber*(1-log2(qber)) + nEdacSets / blockSize;
endfunction

% entropyEstimate -- this applies the specific entropy estimate chosen
% and then turns it into a fraction of the sifted bits.  The entropy
% estimate here is the information Eve may be assumed to have derived
% from eavesdropping on the single-photon pulses, there is a separate
% function for splitting multi-photon pulses.
%
% It can be tricky to compare estimates because of differing assumptions.
% The entropy derived in Bennett's paper (BBBSS92) refers to the entire
% key string, including error bits -- they are kept in the string and
% accounted for as revealed information during error correction.  The other
% estimates derive entropy on the non-error bits.  In these functions,
% we standardize on Eve's entropy on the non-error bits.
%
% We also explicitly subtract the privacy amplification overhead in the
% estimates, since this is different for the Myers-Pearson estimate (it
% uses Renyi order < 2).

function est = entropyEstimate(qber)
    global estType blockSize confidence
    b = blockSize;
    e = qber*b;
    switch (estType)
    case 'Bennett'
        est = bennett(b,e,confidence);
    case 'Slutsky'
        est = slutsky(b,e,confidence);
    case 'Myers'
        est = myers(b,e,confidence);
    otherwise
        error('Unknown entropy estimate type %s',estType);
    end
    est = est/blockSize;
endfunction

function est = bennett(b,e,confidence)
    t = 2.828427*e;
    dev2 = 6.828427*e;
    conf1 = sqrt(2) * erfinv(1-confidence);
    est = b - e - t - conf1*sqrt(dev2);
    est = est + 2*log2(confidence);
```

```
endfunction

function est = slutsky(b,e,confidence)
    conf1 = erfinv(1-confidence);
    eprime = min(e / b + conf1 / sqrt(2*b), 1/3);
    t = (1 - 3*eprime) / (1 - eprime);
    t = (1 + 1.442695*log(1 - 0.5*t*t)) * (b-e);
    dev2 = (b-e)/2;
    est = b - e - t - conf1*sqrt(dev2);
    est = est + 2*log2(confidence);
endfunction

% estimatePNSbits -- how many bits to discard because of "undetectable"
% eavesdropping, i.e. photon-number splitting attacks or unambiguous state
% discrimination (PNS or USD).  This version is essentially Bennett's
% with a more accurate expression for multi-photon pulses.  We assume
% that in all multi-photon pulses, one is captured by Eve and stored until
% the bases are announced.

function mpdisc = estimatePNSbits(sift)
    global mpn detEff0 detEff1 rxLoss
    p0 = exp(-mpn);
    p1 = p0*mpn;
    p2x = 1-p0-p1;
    m = p2x / (p1+p2x);
    mpdisc = m * sift;
endfunction

% estimatePNSgh -- Gilbert & Hamrick's estimate of Eve's information from
% "undetectable" eavesdropping

function mpdisc = estimatePNSgh(sift)
    global fiberLength fiberLoss mpn detEff0 detEff1 rxLoss eveChan
    p0 = exp(-mpn);
    p1 = p0*mpn;
    p2 = p1*mpn/2;
    p2x = 1-p0-p1;
    s2 = sqrt(2);
    y = .1^(.1*(fiberLength*fiberLoss*eveChan + rxLoss)) * (detEff0+detEff1)/2;
    m1 = p2x - 1/(1-y)*(exp(-mpn*y)-exp(-mpn)*(1+mpn*(1-y)));
    m2 = p2*y + 1 - exp(-mpn)*(s2*sinh(mpn/s2)+2*cosh(mpn/s2)-1);
    m3 = p2*y + exp(-mpn)*(sinh(mpn)-s2*sinh(mpn/s2));
    p2k = p2;
    for k = (2:20)
        p2k = p2k * mpn * mpn / (k*(4*k-2));
        m3 = m3 + p2k*max(1-(1-y)^(2*k-1),1-2^(1-k));
    endfor
    m = max([m1,m2,m3]);
    mpdisc = m * sourceRate / 2;
endfunction

% estimatePNSb -- Bennett, et al.'s estimate for Eve's information from
% "undetectable" eavesdropping (BBBSS92)

function mpdisc = estimatePNSb(sift)
    global mpn
    mpdisc = sift*mpn;
endfunction

% estimatePNSx -- a version of our estimate in which we also make allowance
% for USD on 3+ photon pulses.

function mpdisc = estimatePNSx(sift)
    global mpn detEff0 detEff1 rxLoss
    s = sqrt(2);
    h = mpn/s;
    pusd = 1 - exp(-mpn)*(s*sinh(h)+2*cosh(h)-1);
```

```
    p0 = exp(-mpn);
    p1 = p0*mpn;
    p2x = 1-p0-p1;
    mu = p2x / (p1+p2x);
    mpdisc = (mu*sift*(1-pusd)) +
(pusd*sourceRate*(detEff0+detEff1)/2*.1^(.1*rxLoss));
endfunction

% distilledRate -- this is the final answer, number of distilled bits per
% second.

function rate = distilledRate
    global confidence
    [sift, qber] = siftedRate;
    ovhd = EDACoverhead(qber);
    ent = entropyEstimate(qber);
    mpd = estimatePNSbits(sift);
    mpd = mpd + sqrt(2)*erfinv(1-confidence) * sqrt(mpd*(1-mpd/sift));
    rate = max(sift*(ent-ovhd) - mpd, 0);
endfunction

% Myers/Pearson entropy estimate
%
% First we find the probability p for which the first k terms of the binomial
% distribution binom(n,i)*p^i*(1-p)^(n-i) sum up to 'confidence', the
% probability that we're wrong.
%
% Then, given this probability, p, the best conditional probability of Eve
% correctly guessing a bit is:
%
%       pe = .5 + sqrt( p/(1-p) * (1 - p/(1-p)) )
%
% Then Eve's least Renyi entropy (order R) for the n-k non-error bits is:
%
%       h(R) = (n-k)/(1-R) * log2(pe^R + (1-pe)^R)
%
% Now from Cachin's paper (Smooth Entropy and Renyi Entropy), theorem 8,
% we know that the amount of smooth entropy (which we can feed into privacy
% amplification) is at least:
%
%       h(R) - log2(m+1) - r/(R-1) - t - 2
%
% where m-log2(m+1) = n+t, and 2^(-r)+2^(-t) = confidence.
%
% If we ignore the negligible effect of t on the value of log(m), the optimal
% values of r and t are:
%
%       r = log2(R/confidence)
%       t = log2(R/((R-1)*confidence))
%
% and the value of m is approximately:
%
%       m = n + t + log2(n+t+1)
% or    m = n + t + log2(n+t+1+log2(n+t+1+log2(n+t+1)))         etc.
%
% In our internal function, we negate this, so we can minimize.

function h = myers_neg_renyi_entropy (r)
    global myers_n myers_k myers_confidence myers_pe
    h = (myers_n - myers_k) / (1-r) * log2(myers_pe^r + (1-myers_pe)^r);
    t = log2(r/((r-1)*myers_confidence));
    h = h - log2(myers_n+t+1+log2(myers_n+t+1+log2(myers_n+t+1)));
    h = h - log2(r/myers_confidence)/(r-1) - t - 2;
    h = -h;
endfunction

% Another internal function -- the sum of the first myers_k terms of the
```

```
% binomial distribution, minus myers_confidence (so we can find a zero)

function s = myers_binomtail (p)
    global myers_n myers_k myers_confidence
    k1 = myers_k;
    k2 = myers_n-myers_k;
    if (k1 > k2)
        k1 = k2;
        k2 = myers_k;
    endif

    % Compute the highest term, then go backwards

    if (k1*log(myers_n) < 200)
        % exact if < 10^86
        l = 1;
        for i = 1:k1
            l = l * (myers_n-i+1) / i;
        endfor
        t = l * p^myers_k * (1-p)^(myers_n-myers_k);
    else
        % otherwise use Stirling's approximation
        k1 = k1+1;
        k2 = k2+1;
        n1 = myers_n+1;
        l = 1 - .5*log(2*pi);
        l = l + (1/(n1) - 1/(k1) - 1/(k2)) / 12;
        l = l - (1/(n1)^3 - 1/(k1)^3 - 1/(k2)^3) / 360;
        l = l + (1/(n1)^5 - 1/(k1)^5 - 1/(k2)^5) / 1260;
        l = l + (n1-.5)*log(n1) - (k1-.5)*log(k1) - (k2-.5)*log(k2);
        t = exp(l + myers_k*log(p) + (myers_n-myers_k)*log(1-p));
    endif

    % Now loop back to the beginning, but exit if we stop changing sum

    s = t - myers_confidence;
    for k1 = (myers_k-1:-1:0)
        t = t * (k1+1) * (1-p) / (p * (myers_n-k1));
        s1 = s + t;
        if s1 == s
            break
        endif
        s = s1;
    endfor
endfunction

function entropy = myers(n,k,confidence)
    global myers_n myers_k myers_confidence myers_pe

    % Approximate starting point

    p = 1 - InvBetaApprox(n-k,k,confidence);
    myers_n = n;
    myers_k = k;
    myers_confidence = confidence;

    % Solve for probability p, and compute Eve's probability of guessing

    p = fzero('myers_binomtail',p);
    p = min(p,1/3);
    myers_pe = .5 + sqrt( p/(1-p) * (1 - p/(1-p)) );

    % Maximize entropy measure over Renyi order R

    r = fminbnd('myers_neg_renyi_entropy',1.01,2);

    % Return the maximized entropy
```

```
    entropy = -myers_neg_renyi_entropy(r);
endfunction

% Abramowitz and Stegun approximation for the inverse of the incomplete
% Beta function

function v = InvBetaApprox(a,b,p)
    y = sqrt(2) * erfinv(1-2*p);
    l = y*y/6 - .5;
    a1 = 1/(2*a-1);
    b1 = 1/(2*b-1);
    h = 2/(a1+b1);
    w = y*sqrt(h+l)/h - (b1-a1)*(l+5/6-2/(3*h));
    v = a/(a+b*exp(2*w));
endfunction
```

18.3 Calculated Results: Optimal Mean Photon Numbers[39] for a Limited Eve

The introduction to this section is that the general viewpoint we're operating under is that the proofs of security of QKD all live in the world of theory, the actual QKD systems live in the world of devices. There is a list of assumptions that underlies the proofs, and the actual devices may or may not match them. We keep on encountering new ways in which the devices are imperfect. The response can be to make the devices better match the theory, to implement new protocols, or to change the proofs. In this section, we are making certain to allow an Eve whose capabilities are well beyond what devices are capable of now, but not omnipotent (as limited by the theory of quantum physics). Although it is almost heretical to say this, there are many other possible attacks that are more severe and less demanding for Eve than (say) positive operator valued measure (POVM) attacks, even for systems with "unconditional" security proofs (for example, exploiting slight differences in the timing windows of the detectors).

So although most practitioners of quantum cryptography have now converged upon a mean photon number (μ) of 0.1 as a good benchmark value, “contrary to a frequent misconception, there is nothing special about a μ value of 0.1, even though it has been selected by most experimentalists. The optimal value—i.e., the value that yields the highest key exchange rate after distillation—depends on the optical losses in the channel and on assumptions about Eve’s technology.” [Gisin et al.][40]

The most critical factor driving an optimal choice of mean photon number is determining what sort of attacks Eve can employ. For intercept-resend attacks on the single-photon pulses, there is a fairly well-developed theory about how much privacy amplification is necessary. For multi-photon pulses, a number of possible attacks have been proposed and analyzed, but it is by no means clear that the list of possible attacks is complete yet. Many of the theoretically possible attacks are very far from practical implementation with current technology.

Note that these assumptions about Eve’s abilities must be built into the privacy amplification margin used in any working QKD system, so they are by no means idle questions. If one wishes to deploy QKD securely, one must choose these assumptions carefully. Once we have chosen these assumptions and the privacy amplification formula, numerical optimization techniques can determine the optimal multi-photon

[39] This section is derived from D. Pearson and C. Elliott, “On the Optimal Mean Photon Number for Quantum Cryptography,” quant-ph/0403065 v2, 17 May 2004. See that paper for further observations, references, etc.
[40] Just to point out that Gisin and Lutkenhaus papers have different approaches and assumptions to Pearson and Elliott paper.

probability. Therefore it is useful to explicitly list a set of assumed capabilities for Eve for a given scenario, as the rates vary greatly depending on the assumptions.

We must decide, for example, whether we wish to guard against an Eve possessing the capabilities listed in the table below. Many research results assume that Eve possesses all these capabilities; for some papers it is difficult to determine exactly which capabilities are assumed.

Eve Has?	Potential Technological Capabilities for Eve
☑	Perfect detectors
☑	A perfect long-term quantum memory
☑	Adaptive beam-splitters, which split at most *one* photon from the signal
☑	Reliable quantum non-demolition measurement of the total number of photons
	The ability to perform unambiguous state discrimination on pulses with 3 or more photons
	The ability to discriminate multi-photon pulses in intercept/resend attacks
	The ability to substitute low or zero-loss fiber, or to perform quantum teleportation with small loss

It is our belief, following Gisin, et al, that it is reasonable to guard against eavesdropping that is currently feasible, or may be in the not-too-distant future, rather than make deployment infeasible by attempting to guard against theoretical attacks that may never be possible. Note, in particular, that near-perfect detectors, particularly if they can resolve the number of photons in a pulse, adaptive beam-splitters, or quantum non-demolition (QND) measurements can all give us a reliable way to build a true single-photon source, which would, in turn, render PNS attacks harmless. QKD is very likely to shift to true single-photon emitters long before we need to worry about an eavesdropper with a long-term quantum memory. It is one of the greatest virtues of QKD that, unlike classical cryptography, there is no risk that a future powerful adversary endangers our communications in the present.

Accordingly, the check marks in Table 1 indicate which technology we assume Eve has for the purposes of this analysis, and for the current operation of our working QKD systems. We believe that these assumptions are reasonable for current scenarios, since many of the postulated technologies appear to be beyond today's current state of the art.

Finally, given this explicit set of assumptions about Eve's current capabilities, one must select an entropy estimate used as input for privacy amplification. This entropy estimate includes Eve's information from intercept-resend attacks, called by Slutsky et al. the "defense function." Here we use results based on the original entropy estimate in BBBSS92, but our analytic model explicitly calculates three different entropy estimates (BBBSS92, Slutsky, Myers-Pearson). The choice of optimal mean photon number is very similar for all choices of entropy estimate.

Given all these assumptions, we can employ an analytic model (see above) to calculate the optimal mean photon number (μ) over a range of scenarios. Recall that the "optimal" value is that which maximizes the delivery rate of distilled bits / second, i.e., optimizes across the system-wide effects of multi-photon emission probabilities, attenuation, dark noise, sifting, bits revealed during error detection and correction, and the necessary amount of privacy amplification.

The model allows us to extrapolate system performance in a number of scenarios, e.g. if we had longer fibers, a faster pulse rate, or better detectors. In particular, we can analyze the effects of changing the mean photon number. In Figure 18-1 we vary only the mean photon number μ, with all other parameters derived from one of our current QKD systems (with 10.55 km of optical fiber between Alice and Bob). It is very apparent that the current mean photon number μ, approximately 0.1 photon, is far from optimal in this setting. Instead the mean photon number μ should be slightly more than 1 (about 1.15) to achieve the optimal distilled key rate.

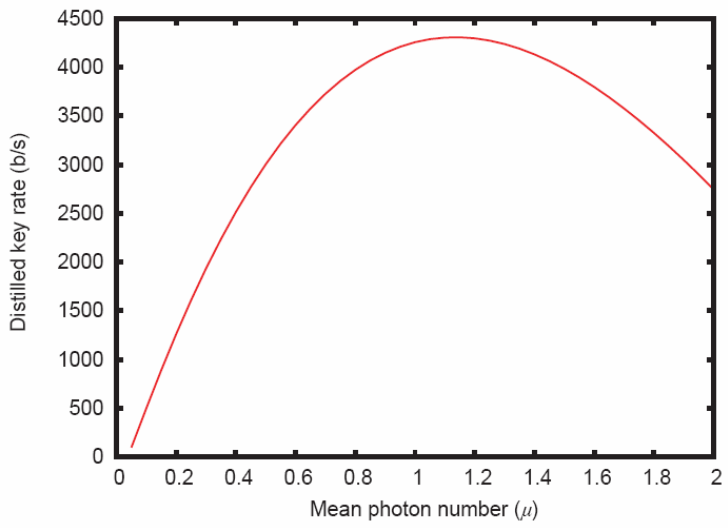


Figure 18-1. Distilled Key Rate as a Function of Mean Photon Number for One Scenario.

Another major objective in optimizing μ is to maximize the distance available for practical QKD over metropolitan fiber. Figure 18-2 shows how the distilled key rate varies with both fiber length and μ, again given specific system characteristics (as above) and the eavesdropping model discussed previously.

As can be seen, the distilled key rate falls off dramatically with distance, and requires high values of μ for long distances. These specific results are driven by the relatively low quantum efficiency, and relatively high dark count, of our current InGaAs detector suite, but the phenomenon is more general. Larger μ naturally leads to more photons at the receiver, and correspondingly more raw key bits per second, but more importantly it keeps the valid detect rate high compared to receiver dark noise. Dark noise with a highly attenuating channel decreases the distilled rate in a very dramatic way because it translates directly into a higher error rate. The error detection and correction protocol, such as Cascade, then must reveal a substantial amount of information to correct the errors. Since it must be assumed, conservatively, that all these errors are due to eavesdropping, the estimate of the remaining entropy in the bits drops sharply.

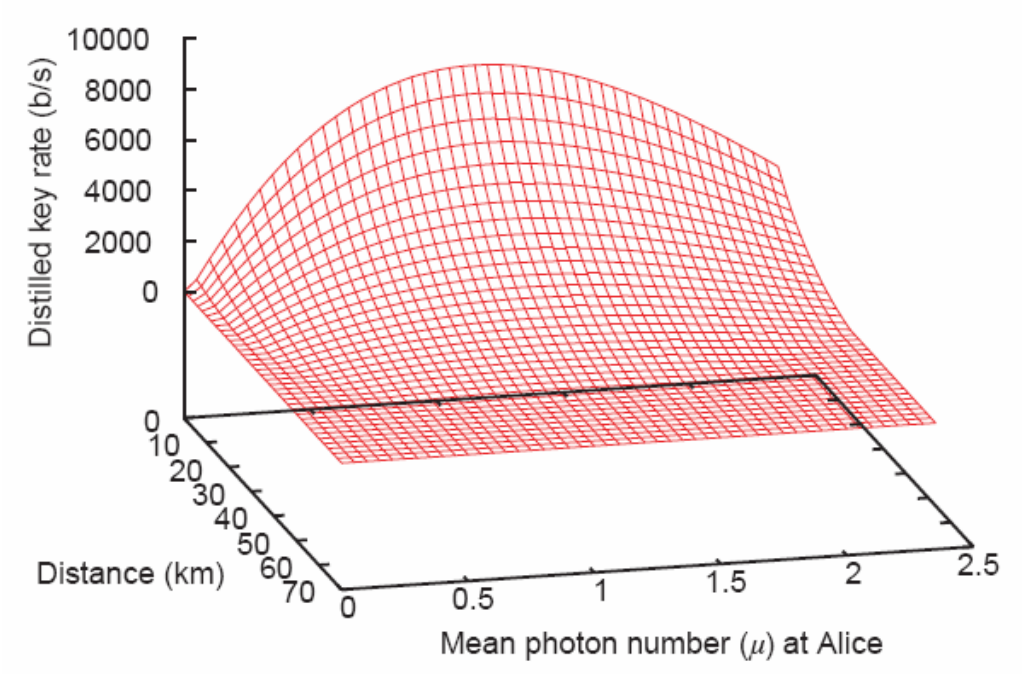


Figure 18-2. Distilled Key Rate as a Function of Mean Photon Number and Distance.

Since many factors affect the distilled key rate, it is not surprising that there is not a single optimum value of μ to employ in all scenarios. However, for our systems, the optimum value does not vary by much. Figure 18-3 shows the optimum μ for distances from zero to 50 km. The optimum varies by less than 20%, from about 1 to 1.2. The peak of the key rate curve (Figure 18-1) is rather broad, so choosing a value of 1.0, say, for μ seems to be applicable for a broad range of operating conditions.

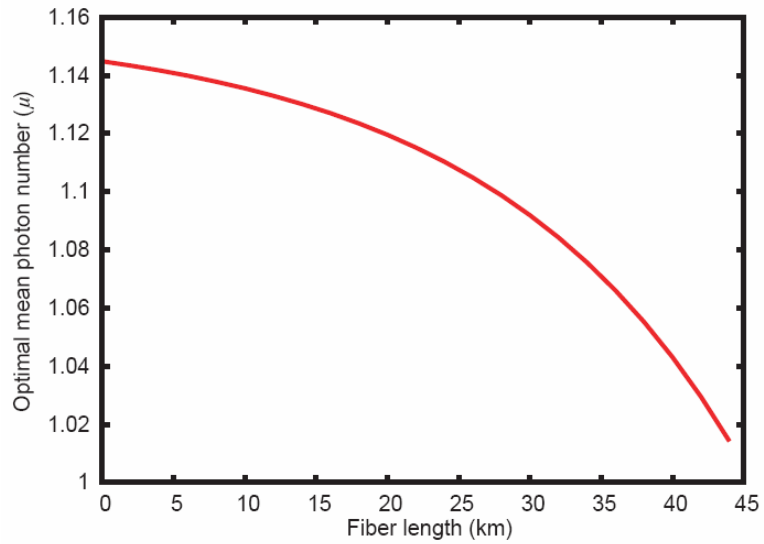


Figure 18-3. Optimal Mean Photon Number as a Function of Distance (Fiber Length).

The threat model treated in this paper has been implicitly assumed in the eavesdropping estimates for multi-photon pulses provided by other research teams. We believe it is a plausible threat model, given current technology. It is, however, important to realize that with larger values of μ we are moving out of the "comfort zone" of these assumptions. Certain attacks that aren't readily feasible at small μ become easier at μ = 1. For example, Bennett et al. considered a special case of unambiguous state discrimination, splitting incoming pulses and measuring one portion in each basis. In some cases of 3 or more photon pulses, the measurement would result in both detectors firing in one basis and one firing in the other. When this happens, Eve can generate a new signal (close to Bob) without introducing any errors. For small values of μ, Bennett et al. concluded this attack was harmless. However, when μ = 1 and with perfect detectors for Eve, this attack becomes feasible with a fiber loss of about 18 dB, corresponding to approximately 90km of fiber at 0.2 dB/km attenuation.

Another attack examined by Gisin et al. involves improving the odds of intercept/resend attacks by splitting the beam, measuring each half in a different basis, and using detectors that can determine the number of photons in the signal. In certain operating regimes (small μ or short fiber length) this attack is no better than traditional intercept/resend, and we may use the same defense function. However by changing the defense function appropriately (i.e. granting Eve more information for each error bit received), one can in fact operate safely with a larger mean photon number. For the operating configuration analyzed in this paper, the result is still an optimal value of μ ≈ 1.1.

18.4 Calculated Results: Throughput Achievable with High-Speed Detectors

As of May 2004, we have employed our analytic model to estimate the effects of a high-speed detector for fiber-based QKD. The target detector resembles a Superconducting Single Photon Detector (SSPD) that we are studying, based on the work of Prof. Roman Sobolewski at Rochester University.

Figure 18-4 estimates the distilled throughput as a function of distance (fiber length), over a range of detector speeds, with fixed 2% Quantum Efficiency (QE) and a fixed, low dark noise.

As one might expect, the resultant throughput is essentially linear with detector speed for much of the operational region. Thus over very short fiber strands, such as within buildings, a 100 MHz detector would provide nearly 10,000 distilled bits per second; a 1 GHz detector would provide nearly 100,000 bits / second; and a 10 GHz detector would provide nearly 1 million bits / second. Over 50 km, these rates fall to about 500 bits/second for a 100 MHz detector; 5,000 bits / second for a 1 GHz detector; and 50,000 bits for a 10 GHz detector.

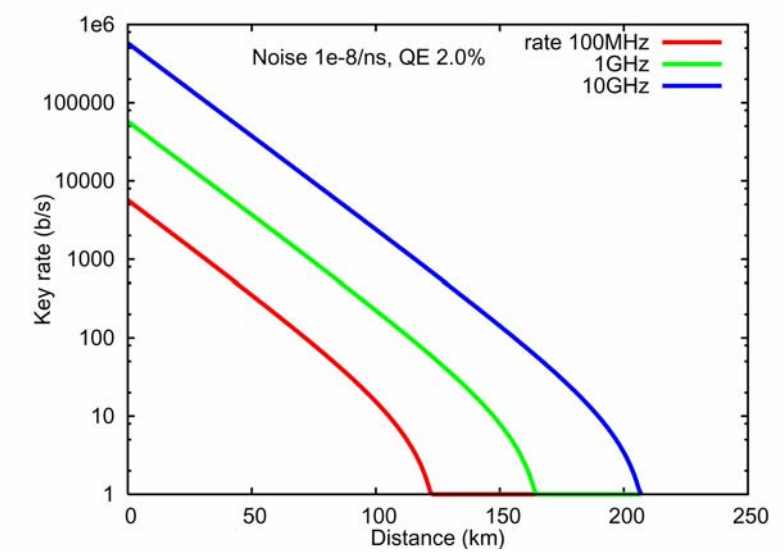


Figure 18-4. Distilled Throughput as a Function of Distance, varying Detector Speeds.

Within these linear operating regions, the throughput is limited by (a) channel attenuation, and (b) the relatively low 2% assumed detector QE. It is thus reasonable to explore the effects of varying the detector QE.

Figure 18-5 estimates the distilled throughput as a function of distance (fiber length), varying the Quantum Efficiency (QE) of a 1 GHz detector with a fixed, low dark noise. Again, the results are not surprising for the linear regions of system operation.

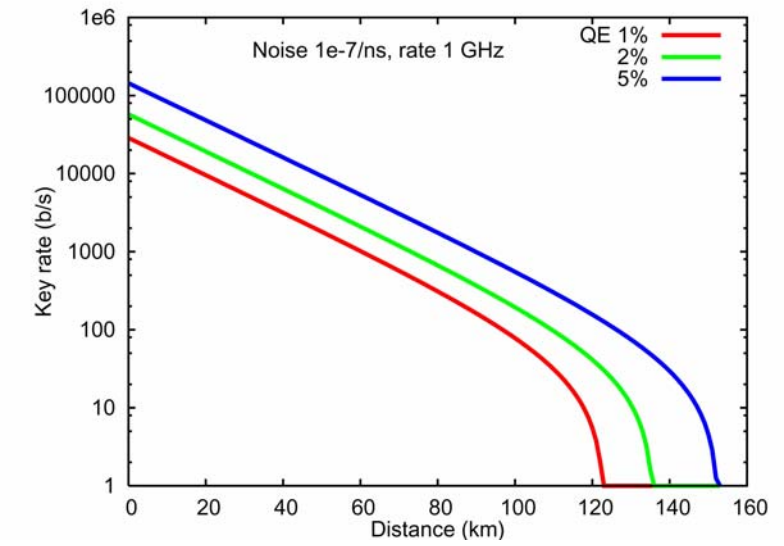


Figure 18-5. Distilled Throughput as a Function of Distance (1 GHz Detector), varying QE.

Over very short fiber strands, such as within buildings, a 1 GHz detector with 1% QE would provide about 20,000 distilled bits per second; with 2% QE it would provide about 40,000 bits / second; and with 5% QE it would about 200,000 bits / second. Over 50 km, these rates fall to about 2,000, 4,000, and 20,000 bits / second respectively.

Figure 18-6 estimates the distilled throughput as a function of distance (fiber length), varying the dark noise characteristics of a 1 GHz detector with 2% Quantum Efficiency (QE). As shown, these low noise rates have virtually no effect on system throughput to about 60 km. After that linear region, they begin to dominate the range at which the system can be operated.

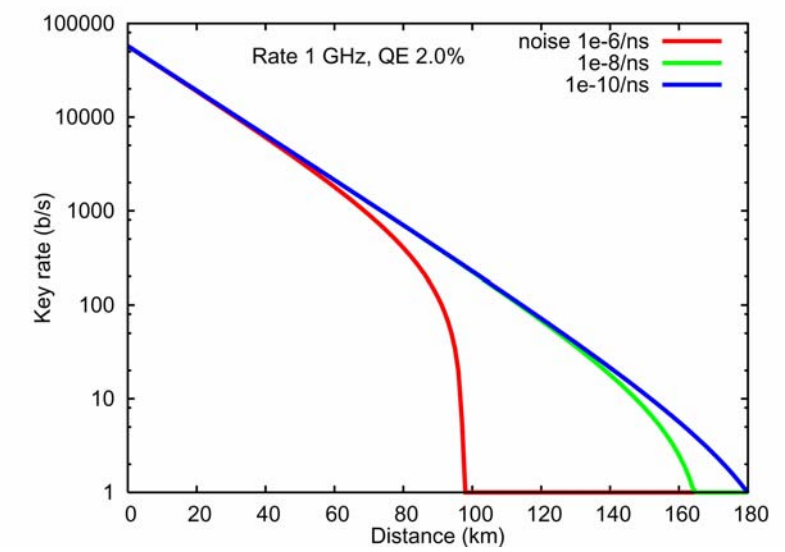


Figure 18-6. Distilled Throughput as a Function of Distance (1 GHz, QE=2%), varying Dark Count.

Attachment A - Notes and Acronyms

This section provides a list of abbreviations and acronyms, with their definitions, as used in this document. It also contains any additional notes needed for this document.

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard, defined in FIPS 197
ALF	Application Level Framing
ANSI	American National Standards Institute
APD	Avalanche Photo Diode
B92	Bennett 1992 2-state QKD protocol
BB84	Bennet-Brassard 1984
BBBSS92	Bennett, Bessette, Brassard, Salvail, and Smolin 1992
BBO	β-Barium Borate
BER	Bit Error Rate
BS	Beam Splitter
BU	Boston University
CORBA	Common Object Request Broker Architecture
CW	Continuous-Wave
CPU	Central Processing Unit
CSPRNG	Cryptographically Secure Pseudorandom Number Generator
DARPA	Defense Advanced Research Projects Agency
dB	Decibel
DES	Data Encryption Standard
DIO	Digital Input/Output
DWDM	Dense Wavelength Division Multiplexing
DSA	Digital Signature Algorithm
EDAC	Error Detection and Correction
E-O	Electro-optical
EPR	Einstein-Rosen-Podolsky
FEC	Forward Error Correction
FIFO	First-In, First-Out
FIPS	Federal Information Processing Standard
FPGA	Field Programmable Gate Array
GF[…]	Galois Field
HMAC	Keyed-Hash Message Authentication Code
Hz	Hertz
IBM	International Business Machines
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
IKE	Internet Key Exchange
ILM	Innamuri, Lütkenhaus, and Mayers
I/O	Input / Output
InGaAs	Indium Gallium Arsenide
IP	Internal Protocol

IPsec	Internet Protocol Security
IPv4	Internet Protocol, Version 4
IPv6	Internet Protocol, Version 6
IV	Initialization Vector
JFK	Just Fast Keying
LFSR	Linear Feedback Shift Register
MAC	Message Authentication Code
Mbps	Megabits per second
MEMS	Micro-Electromechanical System
MHz	Megahertz
MTU	Maximum Transfer Unit
mV	Millivolt
NIM	Nuclear Instrumentation Module, low = ~0.0V and high = -0.8...-1.6V
NIST	National Institute of Standards and Technology
NLC	Non-Linear Crystal
nm	Nanometer
NRNG	Nondeterministic Random Number Generator
ns	Nanosecond
OPC	Optical Process Control
PBS	Polarizing Beam Splitter
PCI	Peripheral Component Interconnect
PFS	Perfect Forward Secrecy
PM	Polarization-Maintaining
PNS	Photon Number Splitting
POVM	Positive Operator Valued Measurement
PRNG	Pseudorandom Number Generator
QBER	Quantum Bit Error Rate
QE	Quantum Efficiency
QKD	Quantum Key Distribution
QND	Quantum Non-Demolition (Measurement)
QPD	QKD Protocol Daemon
QPFS	Quantum Perfect Forward Secrecy
QWDM	Quantum Wavelength Division Multiplexing
RAM	Random Access Memory
RF	Radio Frequency
RFC	Request For Comments
RSA	Rivest, Shamir, Adleman
Rx	Receiver
SA	Security Association
SAD	Security Association Database
SARG02	Scarani, Acin, Ribordy, Gisin 2002 QKD protocol
SHA-1	Secure Hash Algorithm 1, defined in FIPS 180-1
Si	Silicon
SM	Single Mode
SOW	Statement of Work
SPD	Security Policy Database

SPDC	Spontaneous Parametric Down-Conversion
SPCM	Single Photon Counting Module
SSPD	Superconducting Single Photon Detector
TBD	To Be Determined
TBS	To Be Supplied
TCP	Transmission Control Protocol
TDM	Time Division Muliplexing
TEC	Thermo-Electric Cooler
TTL	Transistor-to-Transistor Logic, low = 0...+1V, high = +3...+5V
Tx	Transmitter
UDP	User Datagram Protocol
USD	Unambiguous State Discrimination
VPN	Virtual Private Network
WDM	Wavelength Division Multiplexing
WPRNG	Weak Pseudorandom Number Generator
XOR	Exclusive-Or

Attachment B - Publications, Conferences, Talks

1. J. L. Habif, D. S. Pearson, R. H. Hadfield, R. Schwall, A. J. Miller S. W. Nam, "Single Photon Detector Comparison in a QKD Link Testbed," To be submitted to Applied Physics Letters, 2006
2. Martin Jaspan, Jonathan Habif, Robert Hadfield, and Sae Woo Nam, "Heralding of Telecom Photon Pairs with a Superconducting Single Photon Detector," Submitted to *Applied Physics Letters* January 2006.
3. Gregg Jaeger and Alexander Sergienko "Entangled states in quantum key distribution," Proc. Conf. Quantum Theory, Reconsideration of Foundations - 3, AIP Conf. Proc. 810, 161 (2006).
4. H. E. Brandt and J. M. Myers, "Expanded conclusive eavesdropping in quantum key distribution," *Journal of Modern Optics*, accepted for publication, 2006.
5. J. M. Myers, "Conditional probabilities and density operators in quantum modeling," *Foundations of Physics*, accepted for publication, 2006.
6. C. Elliott, "The DARPA Quantum Network," in *Quantum Communications and Cryptography*, ed. A. Segienko, Marcel Dekker, CRC Press, 2006.
7. C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the DARPA quantum network," *Proceedings of the SPIE,* 5815:138–149, 2005.
8. D. Pearson, C. Elliott, "On the Optimal Mean Photon Number for Quantum Cryptography," forthcoming book chapter.
9. Gregg Jaeger, "Symmetry and concatenated quantum codes," in Quantum information and computation III, edited by E.J. Donkor, A. R. Piritch H. E. Brandt, Proc. SPIE Vo. 5815 (SPIE, Bellingham, WA, 2005), p. 27.
10. Gregg Jaeger, "Entanglement and symmetry in multiple-qubit states: a geometrical approach," Proc. Conf. Foundations of Probability and Physics - 3, Vaxjo, Sweden, AIP Conference Proceedings 750, 180 (2005).
11. Fabio A.. Bovino, Giuseppe Castagnoli, Artur Ekert, Pawel Horodecki, Carolina Moura Alves, and Alexander V. Sergienko, "Direct Measurement of Nonlinear Properties of Bipartite Quantum States," *Physical Review Letters*, v. 95, 240407 (2005).
12. Bahaa E. A. Saleh, Malvin C. Teich, and Alexander V. Sergienko, "Wolf Equations for Two-Photon Light," *Physical Review Letters*, v. 94, 223601 (2005).
13. Ivan Avrutsky and Alexander Sergienko, "Design of Integrated Optical Source of Twin Photons," *Physical Review A*, v. 71, 033812 (2005).
14. F. A. Bovino, P. Varisco, A. Martinoli, P. De Nicolo, S. Bruzzo, A. M. Colla, G. Castagnoli, G. Di Giuseppe, and A. V. Sergienko, "Practical Quantum Key Distribution Using Polarization Entangled States," *International Journal of Quantum Information*, v. 3, pp. 141-146 (2005).
15. F. H. Madjid and J. M. Myers, "Matched detectors as definers of force," *Annals of Physics* **319**, 251–273 (2005).
16. J. M. Myers, "Modeling light entangled in polarization and frequency: case study in quantum cryptography," pp. 147–159 in *Proceedings of the SPIE* **5866**, The Nature of Light: What is a Photon?, C. Roychoudhuri, K. Creath, and A. F. Kracklauer, eds. (SPIE, Bellingham, WA, 2005).

17. T. T. Wu, “Toward a model for multi-qubit quantum memory,” pp. 62–77 in *Proceedings of the SPIE* **5815**, Quantum Information and Computation III, E. J. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2005).

18. J. M. Myers, “Polarization-entangled light for quantum key distribution: how frequency spectrum and energy affect detection statistics,” pp. 13–26 in *Proceedings of the SPIE* **5815**, Quantum Information and Computation III, E. J. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2005).

19. J. M. Myers, “Framework for quantum modeling of fiber-optical networks, Part I, arXiv:quant-ph/0411107 v2 (2005); “Framework for quantum modeling of fiber-optical networks, Part II,” arXiv:quant-ph/0411108 v2 (2005).

20. W. Słysz, M. Węgrzecki, J. Bar, P. Grabiec, M. Górska, C. Latta, V. Zwiller, A. Pearlman, A. Cross, A. Korneev, P. Kouminov, K. Smirnov, B. Voronov, G. Gol’tsman, A. Verevkin, M. Currie, and R. Sobolewski, "Fiber-coupled quantum-communications receiver based on two NbN superconducting single-photon detectors," in: Infrared Photoelectronics, ed. by A. Rogalski, E. L. Dereniak, and F. F. Sizov, SPIE Proc., **5957**, 59570A-1-9, (2005).

21. A. Korneev, O. Minaeva, I. Rubtsova, I. Milostnaya, G. Chulkova, B. Voronov, K. Smirnov, V. Seleznev, G. Gol'tsman, A. Pearlman, W. Slysz, A. Cross, P. Alvarez, A. Verevkin, and R. Sobolewski, "Superconducting single-photon ultrathin NbN film detector," Quantum Electron. **35** No. 8, 698-700 (2005).

22. J. Kitaygorsky, J. Zhang, A. Verevkin, A. Sergeev, A. Korneev, V. Matvienko, P. Kouminov, K. Smirnov, B. Voronov, G. Gol’tsman, and R. Sobolewski, "Origin of Dark Counts in Nanostructured NbN Single-Photon Detectors," IEEE Trans. Appl. Supercon. 15, No. 2, 545-548 (2005).

23. X. Li, Y. Xu, S. Chromik, V. Strbik, P. Odier, D. De Barros, and R. Sobolewski, "Time-Resolved Carrier Dynamics in Hg-Based High Temperature Superconducting Photodetectors," IEEE Trans. Appl. Supercon. 15, No. 2, 622-625 (2005).

24. G. P. Pepe, L. Parlato, R. Latempa, P. D’Acunto, N. Marrocco, C. De Lisio, C. Altucci, G. Peluso, A. Barone, T. Taneda, and R. Sobolewski, "Fabrication and optical properties of ultrathin ferromagnet/superconductor metallic bilayers," IEEE Trans. Appl. Supercon. 15, No. 2, 2942-2945 (2005).

25. G. Gol'tsman, A. Korneev, I. Rubtsova, I. Milostnaya, G. Chulkova, O. Minaeva, K. Smirnov, B. Voronov, W. Slysz, A. Pearlman, A. Verevkin, and R. Sobolewski, (Invited), "Ultrafast superconducting single-photon detectors for near-infrared-wavelength quantum communications," Phys. Stat. Sol. (c) 2, No. 5, pp. 1480-1488 (2005).

26. T. T. Wu, “Quantum cryptography and quantum memory,” pp. 81–92 in *Proceedings of the SPIE* **5436**, Quantum Information and Computation II, E. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2004).

27. J. M. Myers and F. H. Madjid, “Simplified quantum mechanics of light detection for quantum cryptography,” pp. 69–80 in *Proceedings of the SPIE* **5436**, Quantum Information and Computation II, E. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2004).

28. J. M. Myers, T. T. Wu, and D. Pearson, “Entropy estimates for individual attacks on the BB84 protocol for quantum key distribution,” pp. 36–47 in *Proceedings of the SPIE* **5436**, Quantum Information and Computation II, E. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2004).

29. Zachary D. Walton, Alexander V. Sergienko, Bahaa E. A. Saleh, Malvin C. Teich, "Polarization-Entangled Photon Pairs with Arbitrary Joint Spectrum," *Physical Review A*, v. 70, 052317 (2004).

30. Anthony N. Vamivakas, Bahaa E. A. Saleh, Alexander V. Sergienko, and Malvin C. Teich, "Theory of Spontaneous Parametric Downconversion from Photonic Crystals," *Physical Review A*, v. 70, 043810 (2004).

31. Francesco Lissandrin, Bahaa E. A. Saleh, Alexander V. Sergienko, and Malvin C. Teich, "Quantum Theory of Entangled-Photon Photoemission," *Physical Review B*, v. 69, 165317 (2004).

32. Gregg Jaeger, "Bell gems: the Bell basis generalized," Phys. Lett. A 329, 425 (2004)

33. Oleksiy Pikalo, John Schlafer, Alex Colvin, Brig B. Elliott, "Parameter estimation and control in a QKD link," *Proceedings of SPIE*, Volume 5436, pp. 21-27 (2004)

34. C. Elliott, "Quantum Cryptography," *IEEE Security & Privacy*, v. 2, no. 4, pp. 57-61, July/August 2004.

35. J. Myers, T. Wu, D. Pearson, "Entropy estimates for individual attacks on the BB84 protocol for quantum key distribution". *Proceedings of the SPIE* 5436:36–47, 2004.

36. D. Pearson, "High-speed QKD Reconciliation using Forward Error Correction". Proceedings of the 7th International Conference on Quantum Communication, Measurement and Computing (QCMC), pp. 299–302, 2004.

37. C. Elliott, D. Pearson and G. Troxel, "Quantum Cryptography in Practice", Proceedings of the SIGCOMM 2003 Conference, 2003.

38. Brig B.Elliott, Oleksiy Pikalo, John Schlafer, Troxel, Greg "Path-length control in an interferometric QKD link," Proceedings of the SPIE, Volume 5105, pp. 26-38 (2003)

39. G. Di Giuseppe, M. Atature, M. Shaw, A. V. Sergienko, B. E. A. Saleh, M. C. Teich A. J. Miller, S. W. Nam, and J. M. Martinis, "Direct Observation of Photon Pairs at a Single Output Port of a Beam Splitter Interferometer," *Physical Review A*, v. 68, 063817, (2003).

40. Fabio Antonio Bovino, Pietro Varisco, Anna Maria Colla, Giuseppe Castagnoli, Giovanni Di Giuseppe and Alexander V. Sergienko "Effective Fiber-Coupling of Entangled Photons for Quantum Communication," *Optics Communications*, v. 227, pp. 343-348 (2003).

41. Gregg Jaeger, Alexander V. Sergienko, Bahaa E. A. Saleh, Malvin C. Teich "Entanglement, Mixedness, and Spin-Flip Symmetry in Multiple-Qubit System," *Physical Review A*, v. 68, 022318 (2003).

42. Zachary D. Walton, Mark C. Booth, Alexander V. Sergienko, Bahaa E.A. Saleh, Malvin C. Teich "Decoherence-Free Subspaces in Quantum Key Distribution," *Physical Review Letters*, v. 91, 087901 (2003).

43. Aaron J. Miller, Sae Woo Nam, John M. Martinis, and Alexander V. Sergienko, "Demonstration of Low-Noise Near-Infrared Photon Counter With Multiphoton Discrimination," *Applied Physics Letters*, v. 83, 791-793 (2003).

44. Z. D. Walton, A. F. Abouraddy, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich, "One-Way Entangled-Photon Autocompensating Quantum Cryptography," *Physical Review A*, v. 67, 062309 (2003).

45. Zachary D. Walton, Mark C. Booth, Alexander V. Sergienko, Bahaa E.A. Saleh, Malvin C. Teich "Controllable Frequency Entanglement via Auto-Phase-Matched Spontaneous Parametric Down-Conversion," *Physical Review A*, v. 67, 053810 (2003).

46. G. S. Jaeger, M. Teodorescu-Frumosu, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich "Multiphoton Stokes-Parameter Invariant for Entangled States," *Physical Review A* , v. 67, 032307 (2003).

47. G. S. Jaeger, A. V. Sergienko, B. E. A. Saleh, M. C. Teich "Entanglement, mixedness, and spin-flip symmetry in multiple-qubit states," Phys. Rev. A 68, 022318 (2003) .

48. M. Teodorescu-Frumosu and G. S. Jaeger "Quantum Lorentz-group invariants of n-qubit systems," Phys. Rev. A 67, 052305 (2003).

49. A.V. Sergienko and G.S. Jaeger "Quantum information processing and precise optical measurement with entangled-photon pairs," Contemporary Physics 44, 341 (2003).

50. G. S. Jaeger, M. Teodorescu-Frumosu, A. V. Sergienko, B. E. A. Saleh, M. C. Teich, "Multiphoton Stokes-parameter invariant for entangled states," Phys. Rev. A. 67, 032307 (2003).

51. A. V. Sergienko, G. Di Giuseppe, G. Jaeger, B. E. A. Saleh, M. C. Teich, "Quantum metrology and quantum information processing with hyper-entangled quantum state," in Shumovsky and Rupasov (eds.), Quantum communication and information technologies (Kluwer: Dordrecht, 2003), p. 13.

52. G. S. Jaeger, M. Teodorescu-Frumosu, A. V. Sergienko, B. E. A. Saleh, M. C. Teich, "Invariants of multiple-qubit systems under stochastic local operations," Proc. Conf. Foundations of Probability and Physics - 2, Vaxjo, Sweden 2002. Math. Model. Phys. Eng. Cogn. Sci. 5, 273 (2003).

53. T. T. Wu, "Quantum memory: Write, read, reset, and decoherence," pp. 204–215 in *Proceedings of the SPIE* **5105**, Quantum Information and Computation, E. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2003).

54. J. M. Myers and T. T. Wu, "Informed guessing of an eavesdropper's Rényi entropy," pp. 11–18 in *Proceedings of the SPIE* **5105**, Quantum Information and Computation, E. Donkor, A. R. Pirich, H. E. Brandt, eds. (SPIE, Bellingham, WA, 2003).

55. T. T. Wu and M. L. Yu, "Theory and application of Fermi pseudo-potential in one dimension," *J. Math. Phys*. **43**(12), 5949 (2002).

56. J. M. Myers and F. H. Madjid, "Gaps between equations and experiments in quantum cryptography," *J. Opt. B: Quantum Semiclass. Opt*. **4**, S109–S116 (2002).

57. M. Atature, G. Di Giuseppe, M. Shaw, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich, "Multiparameter Entanglement in Quantum Interferometry," *Physical Review A*, v. 66, 023822 (2002).

58. Mark C. Booth, Mete Atature, Giovanni Di Giuseppe, Alexander V. Sergienko, Bahaa E. A. Saleh, Malvin C. Teich, "Counter-propagating entangled photons from a waveguide with periodic nonlinearity,", *Physical Review A*, v. 66, 023815 (2002).

59. G. Di Giuseppe, M. Atature, M. Shaw, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich, "Entangled-Photon Generation from Parametric Down-Conversion in Media with Inhomogeneous Nonlinearity," *Physical Review A*, v. 66, 013801 (2002)

60. M. Atature, G. Di Giuseppe, M. Shaw, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich "Multiparameter Entanglement in Femtosecond Parametric Down Conversion," *Physical Review A*, v. 65, 023808 (2002).
61. A. Abouraddy, A. V. Sergienko, B. E. A. Saleh, and M. C. Teich "Quantum Entanglement and the Two-Photon Stokes Parameters," *Optics Communications,* v. 201, pp.93-98 (2002).
62. A. V. Sergienko, M. Atature, G. Di Giuseppe, G. S. Jaeger, Saleh, B. E. A., M. C. Teich, "Hyper-entangled states and free-space quantum cryptography," Quantum Communication and Information Technologies, Proceedings of SPIE Vol. 4821 (2002), 41.
63. B. Elliott, O. Pikalo, G. Troxel, J. Schlafer, "Path-length control in a interferometric QKD link," SPIE Aerosense 2002, *Proceedings of SPIE* Vol. #5105.
64. D. Pearson and C. Elliott. "QKD Protocols in the DARPA Quantum Network". Proceedings of the 6th International Conference on Quantum Communication, Measurement and Computing (QCMC) 2002.
65. C. Elliott, "Building the Quantum Network," *New Journal of Physics*, v.4, pp. 46.1-46.12, (July 2002).
66. M. Jaspan, *J. L. Habif*, S. Nam, R. Hadfield, "Heralding of Entangled Telecom Photons with a Superconducting Single Photon Detector" Applied Physics Letters, v 89, n 3, 2006, p 031112
67. J. L. Habif*, D. S. Pearson, S. Nam, R. Hadfield, "Single Photon Detector Comparison in a QKD Link Testbed," Proc. SPIE Vol. 63720Z (Oct. 25, 2006).
68. R. H. Hadfield, *J. L. Habif*, J. Schlafer, R. E. Schwall, S. Nam, "Quantum Key Distribution at 1550 nm with Twin Superconducting Single Photon Detectors," Accepted to Applied Physics Letters.

Dissertations

Ph.D. Dissertations from Boston University:

Zachary Walton, "Noise-Immune Entangled-Photon Quantum Cryptography," 2004

MS Thesis from University of Rochester:

Steven E. Rako (M. S. ECE, Nov. 2005) Thesis: "Time-Correlated Single Photon Counting Simulations with Attenuated Pulsed-Laser and Quantum Emitter."

Allen Cross (M. S. ECE, April 2005) Thesis: "Niobium Nitride Superconducting Single-Photon Detectors for Quantum Cryptography."

Public Relations & Press Announcements

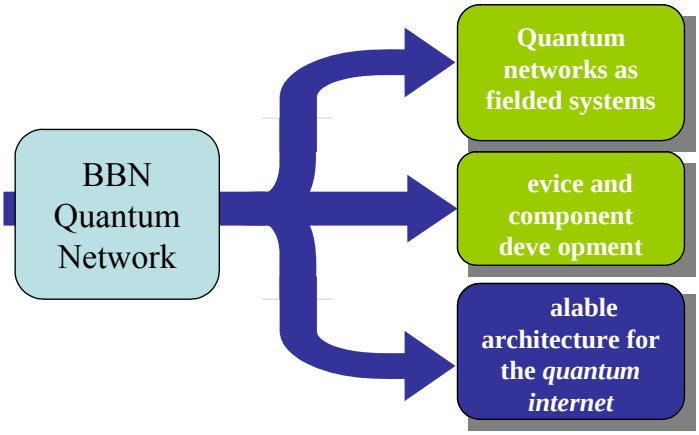
- New Scientist
- Nature
- Government Computing News
- Superconductor Weekly
-

Conferences and Invite talks

- Program Committee for SPIE (quantum)
- ARDA Technical Experts Panel on Quantum Cryptography (writing rev 2.0 of roadmap),
- Co-chaired Naval Studies Board panel on distributed remote sensing,
- Wrote two book chapters on quantum cryptography in 2005, one of which was published this year
- Gave 10-12 universities invited talks during 2005
- Conferences: OFC, Emnets, IEEE LEOS, Asia-Pacific Quantum Info, SIGCOMM
- Invited papers: IEEE LEOS, Asia-Pacific Quantum Info
- Invited talks: U. Montreal
- University Visits in 2006: Dartmouth, Harvard, MIT, Olin, Princeton, Rutgers, UCSB, Washington University

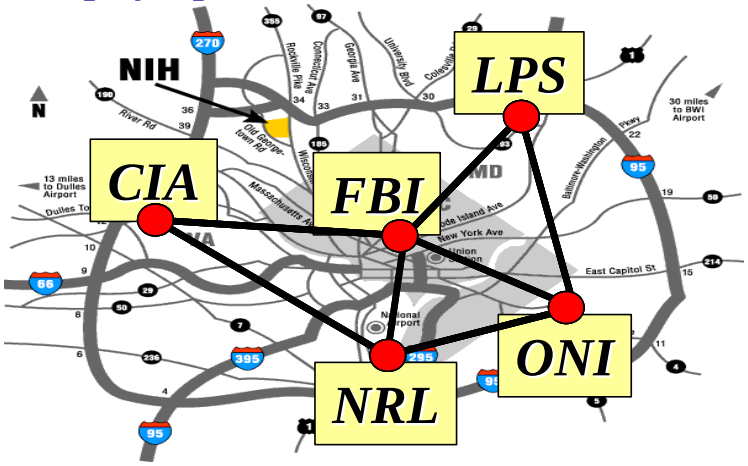
Attachment C – Recommendation on the Future Quantum Communications

It is our view that quantum information is a very fertile ground for technological breakthroughs and rapid fielding of new technologies. This section briefly provides recommendations for "next generation" programs in quantum communications. This attachment highlights our view on quantum information. For more detail description, please refer to a separate document on "*The Future of Quantum Information and Communication.*"



We recommend three different types of follow-on programs, as indicated above. Each has the potential for very high payoff, though the top two have lower risk profiles than the third ("scalable architecture for the quantum internet"). Though quite high risk, this third type of program does have the potential for extremely high payoff.

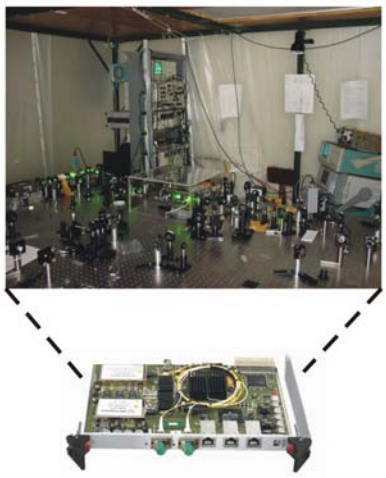
Deploy Operational Network with QKD



First, Quantum Key Distribution (QKD) has now advanced to the stage where it can be used to build the first national QKD. We recommend creation of multiple metropolitan QKD networks linked by long-haul satellite QKD. User enclaves perform rapid rekeying of certified Type 1 devices by quantum

cryptography, and novel protocols enable "routing around Eve" to provide a highly robust key distribution service.

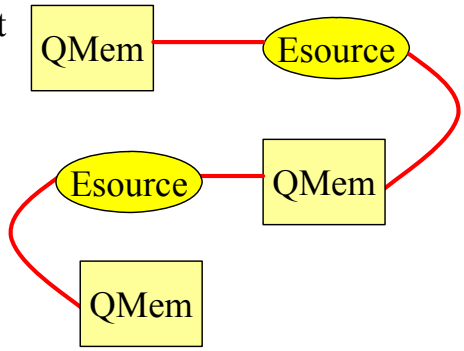
Design and build cost effective quantum components as building blocks for future QIS systems.



Second, progress in quantum information could be greatly accelerated by creation of reliable, inexpensive quantum device technologies that could be economically distributed to multiple research groups. Key technologies in this list include very high-speed, high-fidelity single-photon detectors, robust and high-rate sources of entangled photons, and reliable devices for bridging the light / matter divide (e.g. transferring quantum state from electrons to photons).

Quantum repeaters for a scalable quantum information network

- Entanglement distribution
- Quantum memory
- High speed QKD



Third, it may now be possible to build fully functioning quantum repeaters within a 3-5 year timeframe. If so, it will be possible to be a coast-to-coast Quantum Internet. This is a very high-risk project, but one with almost unimaginable consequences when it succeeds.

To clarify the quantum repeater from section 11.12 the no-cloning law using dim pulses", it may seem that the no-cloning theorem would make a quantum repeater impossible, since there is no way to amplify a quantum signal, or to restore a partly degraded signal. There is, however, a way to increase the fidelity and robustness of a given qubit by using quantum error-correcting codes, and encoding the state in several entangled qubits. Using such codes, it is possible to generate a pair of entangled qubits at a large distance whose fidelity is much higher than could be achieved simply by transmitting a single qubit. One such technique is known as entanglement purification.

Once the coding and decoding have produced a high-fidelity entangled pair, the repeater can use quantum teleportation to send an arbitrary entangled state to the destination -- but the source end of the repeater had to perform a joint measurement on the source qubit and its half of the entangled pair, and in so doing destroyed its copy of the qubit, so there has been no cloning. Further more, the entanglement purification, etc. are components of the quantum repeater. The repeater consists of the following elements:

1) An entangled source (like in the qkd system)
2) A quantum channel (like in qkd)
3) A way to capture the quantum state in a quantum memory
4) A way to perform error detection or correction on several bits in the quantum memory
5) A way to perform quantum teleportation

All of these can be demonstrated but have not been implemented with the same qubit technology -- the main stretch is to interface long-distance qubits (telecomm photons) with a quantum memory.

Using these components, we could generate shared entangled states between two adjacent nodes, verify or improve their fidelity, and store them. Then using quantum teleportation we could reliably transmit a quantum state one hop and store it. This could be extended into a multi-hop store-and-forward network for quantum states. All the operations take place without actually measuring the quantum state we're relaying (otherwise the state would be collapsed). A QKD protocol would work end-to-end through this network with no need to trust the intermediate nodes.